Departamento de Automática y Electrónica

PROYECTO FIN DE CARRERA

INGENIERÍA INDUSTRIAL

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE PARADA DE EMERGENCIA INALÁMBRICO PARA ROBOTS MÓVILES

Autor: HÉCTOR JUÁREZ CALVO

Director: DAVID GARCÍA GODOY

Tutor: RAMÓN IGNACIO BARBER CASTAÑO

Leganés, Octubre 2012

Título: Diseño e implementación de un sistema de parada de emergencia inalámbrico para robots móviles.

Autor: Héctor Juárez Calvo.

Director: David García Godoy.

Tutor: Ramón Ignacio Barber Castaño.

EL TRIBUNAL

Presidente:_____.

Vocal:_____.

Secretario:_____.

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Han transcurrido ya seis años desde que inicié la carrera de Ingeniería Industrial con el objetivo de crecer como persona y como profesional dentro de un campo multidisciplinar, cuya meta es la de tratar de mejorar las condiciones de vida de las personas y crear nuevas fuentes de trabajo. No totalmente consciente de la dificultad de los problemas a los que habría de enfrentarme, en contrapartida, disponía del entusiasmo y las ganas de trabajar interactuando con personas, equipos, materiales, energía e información, siempre con el objetivo en mente de contribuir a la preservación del medio ambiente.

Estos factores supusieron un apoyo, acrecentado por la ayuda y el respaldo de numerosas personas e instituciones que han hecho posible la realización de este proyecto.

Quisiera comenzar reconociendo la paciencia de mis familiares; mis padres, Felipe y Dori y mi hermana Victoria que me han apoyado incondicionalmente a lo largo de todos estos años. A pesar de las dificultades, ellos siempre me han ayudado y me han dado ánimos para terminar lo que empecé.

También quiero agradecer a aquellas personas que ya no están aquí, mis abuelos y mi tía Pilar, que siempre se acordaron de poner una velita cuando tenía un examen difícil y llamaban para preguntar que tal me iba.

A mis compañeros y amigos de la Universidad, con los que he compartido conocimientos, alegrías y momentos de incertidumbre con los duros exámenes. Al final todo se aprueba y se pasa.

Asimismo un reconocimiento a mis amigos de la infancia, que pese a preguntarme infinidad de veces de que va mi proyecto, sin llegar a comprenderlo, siempre me ha felicitado y apoyado a lo largo de todos estos años.

Mi sincera gratitud a la Universidad Carlos III de Madrid, al profesorado y personal de la Universidad, que me han ayudado a adquirir los conocimiento que ahora tengo.

He de hacer especial referencia a mi tutor Ramón Ignacio Barber Castaño y a mi director de proyecto, David García Godoy, que me han facilitado la realización de este proyecto.

Finalmente pido disculpas por la inevitable omisión de algunas personas en esta relación, lo cual no significa que me haya olvidado de ellas.

Resumen

En este Proyecto se desarrolla un sistema de comunicaciones inalámbrico que conecte a uno o varios robots con múltiples mandos de control manejados por diferentes usuarios.

En este sistema prima la seguridad y para conseguir buenos niveles de seguridad es necesario una rápida y eficiente transmisión de datos, en el que el robot debe conocer en tiempo real todas las setas de emergencia asociadas a él y en el caso de una pérdida de comunicación con alguna de ellas, el robot instantáneamente debe pararse.

Además de toda la investigación y diseño del sistema de comunicación, también se ha llevado a cabo su implantación física. Para ello se ha hecho uso de microprocesadores (Arduino FIO) y módulos de radiofrecuencia (Xbee) y se ha desarrollado en lenguaje propio de la plataforma Arduino, pero este lenguaje es idéntico a C.

Finalmente se desarrollaron pruebas que comprobaron el correcto funcionamiento del sistema y también sirvieron para caracterizar el sistema, estableciéndose los tiempos de transmisión de un paquete de datos o duración de la batería del mando de control entre otras características.

Abstract

This Project develops a wireless communication system that connects robots with many remotes control play by many different users.

The most important thing of this system is safety and to get high safety levels is necessary a quick and an efficient communication. Because of this, Robot has to know how many remotes control has associated it and if any of them loose the wireless connection, the system must stop the robot engines.

Beside all research and design of the communication system, I also have built it physically. To make it happen it necessary microprocessors (Arduino Fio) and radio frequency modules (Xbee). The develop language for programming the microprocessors is from Arduino Platform but it is similar to C language.

Finally made tested all the system to observe a proper system behavior and with this tests we can also characterize the communication system, settling down connection times or call duration ion battery and other characteristics.

Índice

Índice de figuras

1. Introducción

El mundo de la robótica no solo ayuda a la industria en los procesos de fabricación sino que también es útil en la vida cotidiana para las personas, por ejemplo un robot aspirador o el Robot Maggie [1], que es donde se centra este proyecto.

Aunque estos robots parezcan inofensivos y poco peligrosos, no nos tenemos que olvidar que siguen siendo máquinas, sobre las que hay que tener un control, y máquinas que en cualquier momento dado tengamos que pararlas, no solo porque puedan causar algún daño físico a una persona, si no por el propio robot, que en algún caso pierda el control y pueda caerse, chocar o impactar con otro objeto pudiendo romperse.

Por esto, para que el robot empiece a moverse y realizar sus tareas programadas, será necesario que alguna persona tenga el dominio sobre él. De esta manara será una persona quien lo active y lo puede parar, todo de forma inalámbrica.

En el caso de Maggie para que el robot entre en funcionamiento será necesario la activación de dos relés, que se activarán por separado, cada uno de forma

independiente, y solo entrará en funcionamiento cuando estos dos relés estén activos a la vez.

También dispone de otro relé que detendrá a Maggie en caso de que se produzca una parada de emergencia. La activación de la parada de emergencia puede activarse de dos maneras diferentes, la primera de ellas es que la base donde se sitúan las baterías y los motores de Maggie se acerque a una cierta distancia de algún objeto, entonces este objeto es captado por unos sensores de proximidad, lo que haría que saltase la emergencia. El otro motivo por el que Maggie podría detenerse es que el usuario presione el botón de emergencia de su mando, este último sistema funciona con un transmisor de radiofrecuencia, similar al que tiene los mandos de apertura de puertas. Estos mandos no siempre son cien por cien fiables, incluso en numerosas ocasiones es necesario apuntar hacia el objetivo para que la señal transmitida llegue correctamente.

Este proyecto trata de la construcción de un sistema de activación y parada de emergencia, en un medio de transmisión inalámbrico que pueda controlar a múltiples robots, mediante el uso de pequeños microprocesadores y módulos de radiofrecuencia XBEE que eliminen cualquier problema de conectividad y haga al sistema de control del robot más seguro.

En definitiva, lo que se quiere implementar es un sistema de comunicaciones. Para definir, desarrollar y establecer las especificaciones del sistema hay que realizar un estudio sobre todas las opciones que el modelo de interconexión de sistema abiertos, también llamado OSI nos ofrece [2].

El modelo OSI fue desarrollado como un modelo de red descriptivo por la ISO, en él se define una normativa formada por siete capas que establece las diferentes fases por las que deben pasar los datos para viajar de un dispositivo a otro sobre una red de comunicaciones. Este modelo de siete capas viene representado en la Figura 1.

Figura 1: Pila OSI.

En este proyecto se ha realizado el desarrollo del sistema de comunicaciones la capa número dos de la pila OSI, el nivel de enlace de datos. Esta capa es responsable de la transferencia fiable de información a través de la red de comunicación. La capa de enlace de datos hace uso del nivel físico. El nivel físico es donde se definen los medios físicos donde se realiza la comunicación, en el sistema de comunicación implementado será el air ya que es un sistema de comunicación inalámbrico. Además aquí se definen otros parámetros físicos como las características materiales, componentes, conectores, niveles de

tensión que se usan durante la transmisión. En el proyecto el componente fundamental que impone estos parámetros es el módulo de radiofrecuencia utilizado, que más adelante se explicará.

El nivel de enlace de datos está un escalón por encima del nivel físico, cuyo objetivo será que la información fluya sin errores entre dos dispositivos. Para que todo ello sea posible se deben crear unos protocolos que inicien y terminen la comunicación de manera segura, segmenten los paquetes para poder realizar controles del flujo de información y controles de errores de manera más rápida, y en caso de que se produzca un error, el sistema debe saber que se ha producido el fallo.

1.1. Motivación

El sistema usado hasta ahora consistía en simples controles que se comunicaban por radiofrecuencia tradicional. Esto supone un problema, ya que puede haber interferencias físicas (obstáculos y objetos intermedios) que obstaculicen la recepción de un señal de emergencia desde una seta para la detención del robot, así como también podemos llegar a perder el control del robot si la seta de emergencia esta demasiado alejada del robot, pudiendo ser en ambos casos una situación peligrosa tanto para el robot como para las personas y objetos del medio donde se trabaje.

Por ello con el nuevo sistema que se implanta, se trata de que estas situaciones no sucedan, mejorando y haciendo que el sistema sea más seguro.

1.2. Objetivo del proyecto

El objetivo de este proyecto es:

- **Implementación de un nuevo sistema inalámbrico de arranque y parada de emergencia para robots.**

Este objetivo se puede desglosar en:

- Implementación física de la seta de emergencia.
- Programación de la seta de emergencia.
- Pruebas de comunicación.

El principal objetivo del proyecto es la sustitución del antiguo sistema de arranque y parada de emergencia para un robot por otro sistema más sofisticado. Para ello se implementará un sistema de comunicación entre setas y robot en el que siempre estarán en contacto, es decir se creará una red de comunicación continua, en el que el robot pueda conocer el número de setas que hasta entonces había y así, si pierde la comunicación con alguna de las setas, el sistema se daría cuenta y hará que el robot entre emergencia por motivos de seguridad, ya que para un usuario el robot estaría fuera de su control.

Asimismo el usuario podrá realizar una parada de emergencia controlada pulsando el interruptor de parada de la seta, deteniendo de esta manera el robot.

Todo ello requerirá de un estudio de los principales sistemas de comunicación inalámbrica y de una búsqueda de información sobre como se puede implementar el sistema.

Finalmente se deberán realizar las pruebes pertinentes para que el sistema funcione correctamente.

1.3. Partes del documento

Una vez realizada una breve introducción del proyecto, pasaremos a explicar el estado del arte y el marco tecnológico así como todos los materiales usados para la realización del sistema, así como del diseño físico y programación tanto de la parte implementada en la seta como en la situada en el robot.

También será necesario la realización de experimentos para controlar los tiempos de reacción desde que el usuario da una orden de parar hasta que el robot se para, ya que el tiempo es una variable crítica si se habla de una emergencia.

Finalmente se expondrán las conclusiones y posibles mejoras del sistema para un caso futuro.

Todo ello se desarrollara en los siguientes capítulos de este documento:

En el capítulo 2 se realiza una descripción de todo el hardware utilizado en el proyecto, justificando su uso.

En el siguiente capítulo, el capítulo 3, se desarrolla una introducción y una breve explicación sobre el software de apoyo para la programación del microprocesador y de los módulos de radiofrecuencia.

A continuación, en el capítulo 4, se explicará el sistema alojado en el robot; se especificarán todos sus componentes usados y se estudiará su funcionamiento lógico gracias a la ayuda de los diagramas de flujo.

En el siguiente capítulo, el capítulo 5, se realizará el mismo estudio que se comentó en el anterior capítulo, únicamente que se centrará en el estudio del sistema alojado en la seta de emergencia.

En el capítulo 6 se terminará de explicar todas las partes del sistema por separado. En este capítulo se explicará como se realizó la carcasa para la seta de emergencia.

En el capitulo 7 se detalla el funcionamiento del sistema en su conjunto formado por la seta y el robot, así como se explicarán todas las pruebas realizadas y los resultados de dichas pruebas.

Una vez terminada la explicación de los test, se comentarán las conclusiones sobre el proyecto y además se justificarán trabajos futuros para la mejora del sistema en el capítulo 8.

En el último capítulo de la memoria, el capítulo 9, se realiza una estimación del coste total de todo el sistema para poder realizar el presupuesto final.

Finalmente, en los anexos, se exponen las hojas de características de los componentes usados, el manual de configuración de los módulos de radiofrecuencia, algunos de los datos obtenidos en las pruebas y el código desarrollado tanto en el sistema alojado en el robot como en el sistema alojado en la seta de emergencia.

2. Descripción del Hardware utilizado en el sistema

El nuevo sistema de puesta en marcha y parada del robot requiere el uso de un microprocesador para su implementación, ya que es necesario que seta y robot estén en continua comunicación. El uso de electrónica analógica y radiofrecuencia convencional RF haría que este proceso sea complicado, e incluso difícil de conseguir, en cambio con el uso de un microprocesador junto con un módulo de radiofrecuencia que opere en el rango de frecuencias libres, hace que la programación y el sistema sea más sencillo, potente y seguro, ya que permite implementar protocolos de respuesta y que los mensajes lleguen correctamente, además permitiría enviar más información y no únicamente

señales de puesta en marcha y parada, que es la única función que realizaba el sistema anterior.

Por otro lado se utilizarán las entradas analógicas del microprocesador para saber la carga de la batería y mediante salidas digitales se visualizaran en LEDs el estado de conexión con el robot, el estado del robot una vez conectado y la carga de batería de la seta.
A continuación se explica más detalladamente cada componente (para más información ver anexo A1).

2.1. Microprocesador

En el mercado existe una amplia gama de microprocesadores [3], de diverso tamaño, potencia, memoria y precio.
El mejor microprocesador para esta aplicación es de la familia "ARDUINO" [4]. Arduino es una plataforma "open – hardware" de diseño y distribución libre, compuesta por una pequeña placa con entradas y salidas analógicas y digitales.
Existen múltiples placas Arduino, pero la opción que mejor se adapta al proyecto es el "ARDUINO FIO" (*Figura 2*), y se ha elegido este microprocesador debido su pequeño tamaño, se puede incorporar directamente un módulo de radiofrecuencia sin hardware intermedio, se puede conectar directamente una batería LYPO e incorpora el sistema de carga y todo ello con un precio bajo.

Este microprocesador no es el más potente del mercado, pero para la aplicación, tanto la memoria como el cristal es más que suficiente. A continuación se presenta una imagen del "ARDUINO FIO".

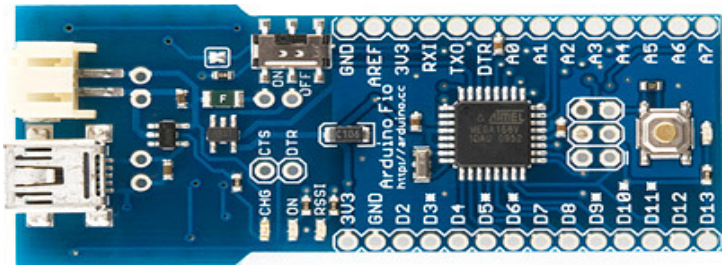


Figura 2: Microprocesador Arduino Fio.

Aunque la plataforma de programación así como el lenguaje de programación para este microprocesador es propio de ARDUINO, este tiene grandes similitudes con el lenguaje de programación C, en el que se pueden incluir librerías propias de lenguaje C [5] (como por ejemplo *stdio.h*), un lenguaje sencillo y muy usado, por lo que no hay ningún problema extra en la programación de este microprocesador [6].

El programa para programar se denomina ARDUINO ALPHA (*Figura 3*) y es un programa OPEN, al igual que toda la plataforma ARDUINO, lo cual da cierta libertad a la hora de programar.

En el propio ARDUINO ALPHA se puede escribir el código, cargar las librerías a utilizar por el programa, compilar el programa y cargarlo en el microprocesador. Así mismo ARDUINO ALPHA posee un terminal para comunicación serie con la placa.

Figura 3: Aspecto del programa ARDUINO ALPHA

Una pequeña complicación surge cuando hay que cargar los programas en el microprocesador, ya que la conexión USB que tiene la placa ARDUINO FIO se usa únicamente para cargar la batería del microprocesador. Aun así existen tres alternativas para cargar los programas:

- Usar un cable FTDI (*Figura 4*): cable que se conecta directamente a los pines RX, TX, alimentación de 3.3 voltios, tierra y tensión de referencia.
- Usar un adaptador conectado a los pines de FTDI (*Figura 5*): este sistema de transmisión de datos utiliza el mismo principio que el anterior, únicamente que éste permite usar cables micro USB en vez del cable FTDI.
- Usar adaptador inalámbrico: es el método que se ha usado en el proyecto para cargar los programas en la placa FIO. Se requiere dos módulos de radiofrecuencia XBEE (*Figura 6*), que más adelante se explicarán con más detalle ya que son los que se usan para establecer la comunicación entre la seta de emergencia y el robot.

  También será necesario un adaptador XBEE a USB, denominado XBEE EXPLORER (*Figura 7*). Configurando debidamente estos módulos con ayuda del programa X-CTU se podrán cargar tanto los códigos de programa en el microprocesador, como establecer una comunicación serie con el terminal del ARDUINO ALPHA del ordenador (más adelante

se explica como se configura los módulos XBEE y como funciona X-CTU) o establecer la comunicación entre la seta y el robot.

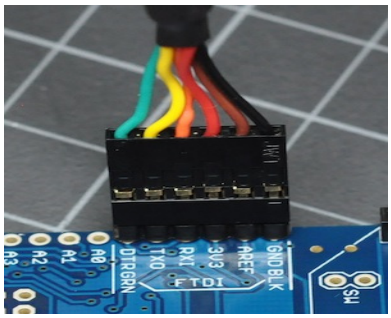


Figura 4: Conexión del cable FTDI con la placa ARDUINO FIO.

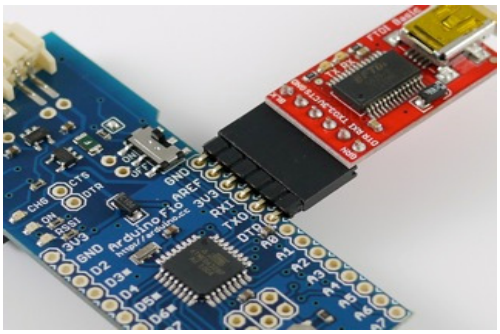


Figura 5 : Conexión con el adaptador micro USB.

Figura 6: Módulo de radiofrecuencia XBEE PRO

Figura 7: XBEE EXPLORER, convertidor de XBEE a USB.

2.2. Batería

La batería (*Figura 8*) utilizada es de tipo LYPO (Polymer Lithium) de dos celdas y proporciona una tensión de 3.7 V entregando un flujo de corriente de 900 mAh.

Es una fuente de alimentación ligera (únicamente 18.5 g) y eficiente, lo que es una característica fundamental a la hora de incorporarlo a la seta inalámbrica. Además posee el mismo conector JST que tiene la placa ARDUINO FIO, por lo que la conexión con la misma es sencilla.

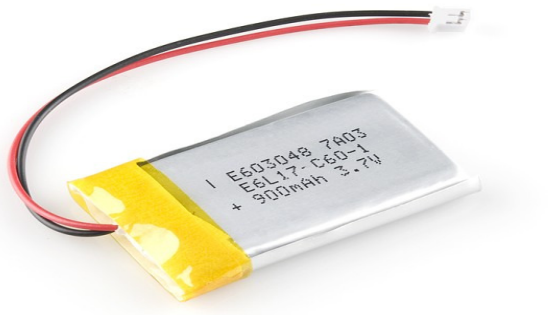


Figura 8: Bateria LYPO de 3.7 V y 900 mAh.

Además la batería trae incorporada la electrónica necesaria para que la tensión entregada por la batería sea lo más próxima a la nominal, a pesar de los posibles picos de corriente que pudieran perturbar el comportamiento de la batería.

Asimismo hace que proceso de carga de la batería sea lo más seguro posible, de manera que cuando la batería se haya cargado completamente no se sigue inyectado corriente a la misma.

2.3. Módulo de Radiofrecuencia

El componente para transmitir la información que más se adecúa a la aplicación es un módulo de radiofrecuencia de la familia de "Digi International" y fabricante "MaxStream", ya que la principal ventaja que poseen estos módulos es que se pueden conectar directamente al microprocesador ARDUINO FIO (es "plug and play"), además funciona a 3.3 V, que es la misma tensión de funcionamiento que la placa.

"MaxStream" suministra una gran diversidad de chips para la comunicación inalámbrica, con diferentes tecnologías y frecuencias de transmisión.

A la hora de elegir este componente se barajaron dos tecnologías parecidas, una es la tecnología XBEE SERIE 1 y la otra es la XBEE ZB ZigBee SERIE 2 (*Figura 10*). Ambas siguen el protocolo 802.15.4 para la comunicación punto a punto y punto a multipunto, el cual define una serie de protocolos de comunicación, a una velocidad de transmisión relativamente baja para

redes inalámbricas personales de corto alcance (WPANs, wireless personal area networks) [7].

XBee opera a 868MHz (Europa), 915MHz (América) y 2.4GHz (Mundo). La velocidad de transmisión máxima es de 250 K bits por segundo.

En la siguiente figura se enmarca la red utilizada dentro de la gran variedad de tipos de redes [8].

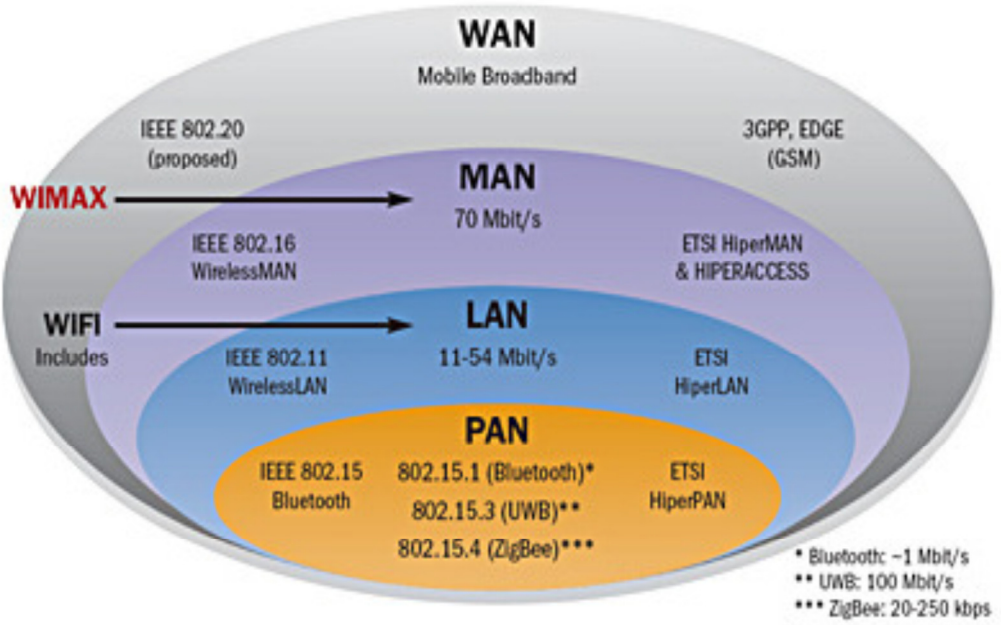


Figura 9: Tecnologías de redes inalámbricas.

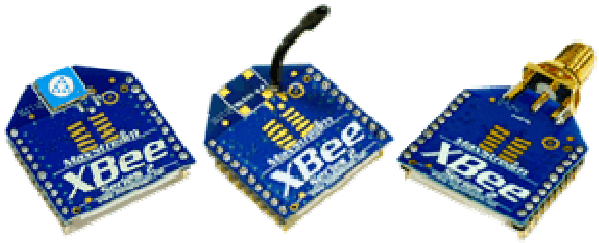


Figura 10: Familia de módulos de radiofrecuencia de MaxStream.

La tecnología ZigBee es más nueva y avanzada y permite crear redes "mesh" (*Figura 11*). Eta red es similar a la red que posee internet, en el que serían necesario una red de "routers", "coordinadores" y "end device" para poder hacer funcionar correctamente a esta red.

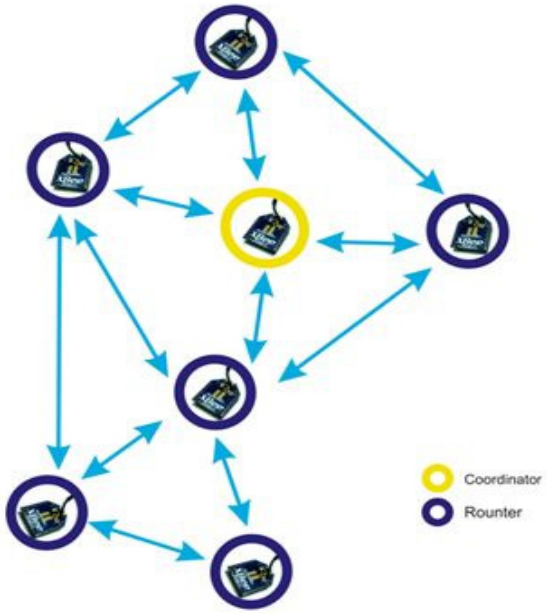


Figura 11: Red Mesh

La opción elegida para la aplicación es el XBEE SERIE 1. Este módulo permite crear una comunicación en estrella (*Figura 12*), que es la usada a la hora de comunicar el robot con los interruptores de emergencia. En esta topología el robot se sitúa como nodo central o coordinador y las "setas" como "end device".

Figura 12: Comunicación en estrella.

Dentro del XBEE SERIE 1 (*Figura 13*) se ha elegido la versión PRO ya que es más potente y permite mayor distancia en la comunicación y finalmente se ha elegido una frecuencia de trabajo 2.4 GHz, ya que es una banda de frecuencia libre.

Figura 13: Modulo XBEE SERIE 1 PRO.

Estos módulos deben tener un pre configuración realizada con la ayuda del software X-CTU que se explicara más adelante y mediante el "XBEE EXPLORER" (*Figura 7*) mencionado anteriormente.

2.4. Otros componentes

Hasta ahora se han comentado y explicado los componentes más relevantes utilizados en este proyecto, pero para que todo el sistema en su conjunto funcione correctamente y tenga sentido han sido necesarios otra serie de componentes que se describen a continuación:

- Pulsadores: será necesario un pulsador para la parada de emergencia, dos pulsadores para el arranque (Figura 14).

- Sensor táctil para conocer el nivel de carga de la batería (*Figura 15*).

- Interruptor: para apagar y encender la seta (Figura 16).

- Switch: este componente con el que interactúa el usuario para la elección del robot al que se desea controlar (Figura 17).

- LEDs: se utilizaran leds de 3mm y 1.8V de bajo consumo, los colores serán verde para saber la conexión o no con el robot (*Figura 18*), azul para saber el estado de funcionamiento del robot y tres leds amarillos para el estado de la batería. Además se pondrán led del color de cada pulsador en el robot, en serie con los relés para saber de manera cierta el botón pulsado

- Resistencias: serán necesarias resistencias de 10k para la conexión de los pulsadores y otras resistencias de 37 ohmios para la conexión de los leds.

- Transistor: será necesario un array de transistores para poder enclavar los relés que activan y desactiva al robot, ya que con la corriente que entrega los pines del microprocesador no es suficiente para activarlos. El transistor elegido es el ULN2065B de STMICROELECTRONICS (*Figura 19*).

Figura 14: Pulsador para la emergencia.

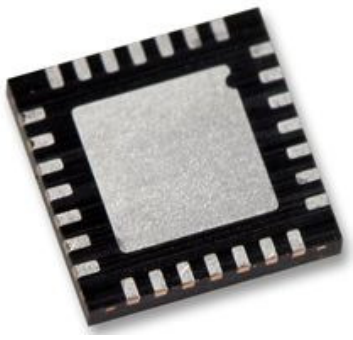


Figura 15: Sensor táctil para conocer la carga de la batería.

Figura 16: Interruptor de encendido de una seta de emergencia.

Figura 17: Switch de elección de robot.

Figura 18: Led verde.

Figura 19: Transistor ULN 2065B de STMICROELECTRONICS

3. Aplicaciones Software utilizadas

3.1. Software de configuración del módulo XBEE

Para que el sistema funcione correctamente y que los paquetes se envíen a los destinatarios deseados es necesario realizar una preconfiguración de los módulos de radiofrecuencia XBEE.

Para ello se utilizará el software X-CTU [9] (*Figura 20*) junto con el explorer de XBEE (*Figura 7*). Este software de Digi International está destinado únicamente a la configuración de chips de radiofrecuencia de toda la familia de MaxStream.

Con este software además de podrá comprobar y visualizar la comunicación entre ordenador y Xbee ya que incorpora un terminal donde pueden

enviarse dígitos codificados en código ASCII o en hexadecimal. También a su vez se puede visualizar la respuesta enviada por el módulo Xbee, siempre y cuando la configuración del emisor y receptor sea la adecuada.

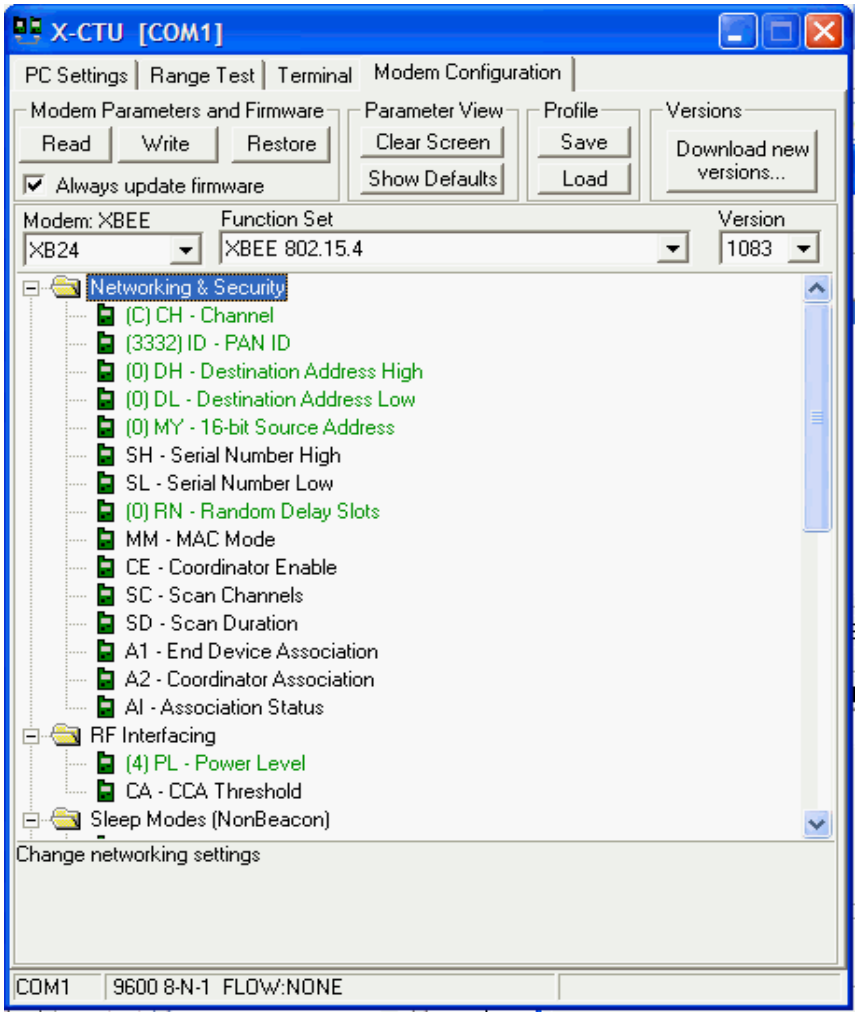


Figura 20: Aspecto del programa X-CTU.

La transmisión del código al microprocesador se realizará también de forma inalámbrica, por lo que tendremos que configurar los módulos de manera diferente:

- Configuración XBEE para transmisión del código: Para este apartado se configurara los módulos XBEE como un simple puerto serie, por lo que únicamente será necesario grabar la velocidad de transmisión, que en este caso será de 57600 baudios y las direcciones de destino (Ver anexo A2).

- Configuración XBEE para transmisión de los paquetes en la aplicación: En la aplicación no se puede utilizar una simple comunicación serie, ya que debido a la topología de red necesaria (estrella) tendremos una transmisión de datos punto-multipunto. Además se utiliza las librerías para el uso de los módulos XBEE, y para ello deberemos activar el modo API de los chips, esto permite crear la red deseada. Además en el caso de que sea la placa para el robot, deberemos configurar ese módulo como coordinador, ya que por él pasarán todo los paquetes.
  El resto de setas se configuraran como end device.

  También será necesario grabar el identificador de red (PAN ID) de cada robot (cada robot creará una red diferente, por lo que cada robot se configurará con PAN ID distintas) y su dirección (source Addres). En el caso de que sea una seta solo será necesario especificar la dirección (cada seta tendrá una dirección diferente, ya que la dirección actuará como identificador de la seta), ya que la seta se conectará a diferentes PAN ID, dependiendo del robot que se desea controlar (ver anexo A3).

3.2. Entorno de desarrollo de aplicación

Como se explicó en apartados anteriores, el software que se utiliza para programar los microprocesadores de Arduino es el ARDUINO ALPHA (*Figura 3*).
Este software de licencia libre, es un programa sencillo de usar pero a su vez completo, ya que incluye librerías, ayudas y ejemplos para las librerías.

También dispone de un terminal para poder visualizar los paquetes enviados y recibidos cuando se establece una comunicación serie con dispositivos externos.

El lenguaje de programación que se utiliza es propio de Arduino cuya extensión del archivo formado al crear un sketch o programa es .ino, pero este lenguaje es similar a C, pese a no tener la extensión .c.
Inclusive tiene la posibilidad de utilizar todas las librerías con extensión .h de C. (en la aplicación se usará la librería xbee.h con el que se puede manejar y controlar fácilmente la gestión de los paquetes enviados y recibidos a través del módulo de radiofrecuencia).

4. Sistema alojado en el robot

El dispositivo alojado robot forma el nodo central de nuestro sistema de comunicación, por lo que actuará como coordinador, es decir por él pasarán los paquetes enviados por todas las setas del sistema y este nodo determinará cual es la siguiente ejecución que realizará el sistema.

Sus funciones principales son la de la recepción de los paquetes, análisis de los mismos y actuación sobre el robot para ponerlo en funcionamiento o pararlo en función de los pulsadores seleccionados en la seta de emergencia.

4.1. Desarrollo Hardware

Los componentes más importantes para que la comunicación se establezca correctamente y que cada seta se comunique con el robot deseado, es el módulo de radiofrecuencia Xbee junto con el microprocesador que es quien coordina dichas comunicaciones y realiza las operación que modificarán el estado del robot.

Cada robot creará una red diferente, es decir cada red vendrá definida con un identificador de red distinto de un robot a otro (PAN ID), pero la dirección de cada robot será la misma, por lo que las setas deben cambiar la PAN ID a la que se conectarán, pero siempre enviarán los paquetes a la misma dirección. Para realizar estos ajustes sobre el sistema alojado en el robot será necesario configurar los módulos de radiofrecuencia Xbee con el programa X-CTU.

Por otra parte, este sistema debe modificar el estado del robot dependiendo del tipo de mensaje recibido, bien poniéndolo en funcionamiento o parándolo. Para ello será necesario actuar sobre unos relés que alimentan a los motores del robot. Se dispondrá de tres relés, uno para la parada de emergencia y los otros dos para la activación del robot. Se colocarán estos dos relés de activación por una cuestión de seguridad, ya que de manera accidental se podría activar un solo botón, por eso hasta que no se presionan los dos a la vez el robot no entraría en funcionamiento.

Para realizar la activación de estos relés será necesario que se realice a través de un transistor bipolar, que actuará como interruptor, ya que la corriente proporcionada por los pines de salida del microprocesador es insuficiente para que los relés se enclaven [10] [11].

La alimentación de corriente de este sistema no será con una batería LIPO de 3.7 V como las de la seta de emergencia, sino que la proporcionará la batería que alimenta al robot. Esta es de 24 voltios, pero posee salidas con una tensión regulada de 5 voltios, la cuál está dentro de los márgenes de tensión a la que se puede alimentar al Arduino Fio. Además al usar la batería del robot evitamos tener que introducir conexiones extra para la carga de una batería LiPO de 3.7 V.

Se puede visualizar el esquema eléctrico de activación y parada del robot y todas las conexiones realizadas con el Arduino Fio en la Figura 21:

- En la zona de la izquierda de la imagen está representado el microprocesador Arduino FIO.
- De las salidas digitales del Arduino se conectan tres LEDs de colores diferentes para poder visualizar que el sistema alojado en el robot a recibido un paquete que modificaría el estado del robot.
- En serie con los LEDs se puede observar las conexiones realizadas al transistor bipolar.
- Finalmente, en la parte de la derecha de la imagen se visualizan los relés de activación de y parada del robot.

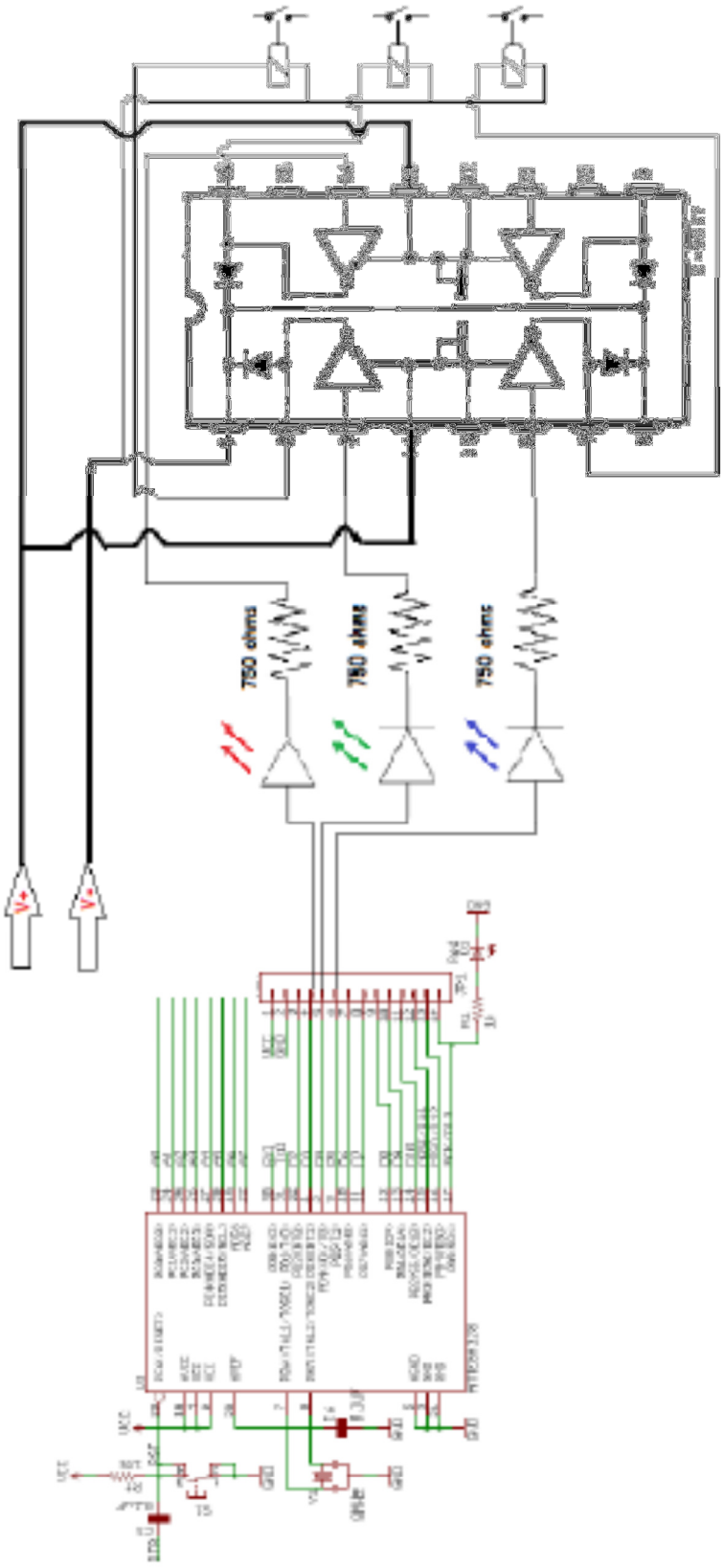


Figura 21: Esquema electrónico de conexiones del sistema del robot

Asimismo todas las conexiones de los componentes externos al microprocesador se alojan en una placa en la que todas las uniones irán soldadas a dicha placa con estaño.

Para unir la placa con el microprocesador se utilizarán conexiones de contactos y el cableado de los interruptores, pulsadores y LEDs se realizará con conectores molex (Figura 22).

Es importante que en caso de que este conector se suelte, sea el conector hembra la que proporcione tensión y el macho el que reciba esta tensión ya que se podría establecer un cortocircuito indeseado, perturbando el sistema o incluso pudiendo estropear algún componente.

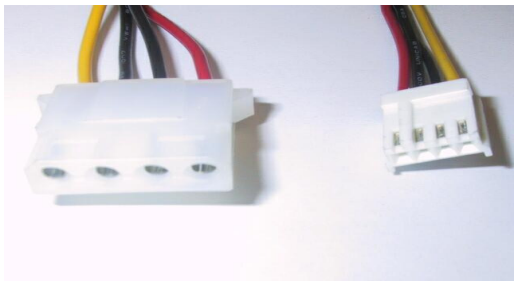


Figura 22: Conector molex.

Hay que añadir que a este sistema se le instalado una nueva conexión serie alojada en dos de los pines digitales, ya que se necesitan dos puertos TX y RX para la transmisión de datos y tierra para la referencia de tensión. Este puerto se ha dispuesto para una posible futura ampliación, ya que a través de éste puerto serie extra, el robot se podría comunicar con el sistema de parada de los motores y con las setas de emergencia de manera sencilla.

En la Figura 23 se puede visualizar una imagen explicativa con el sistema que se implantó en Maggie.

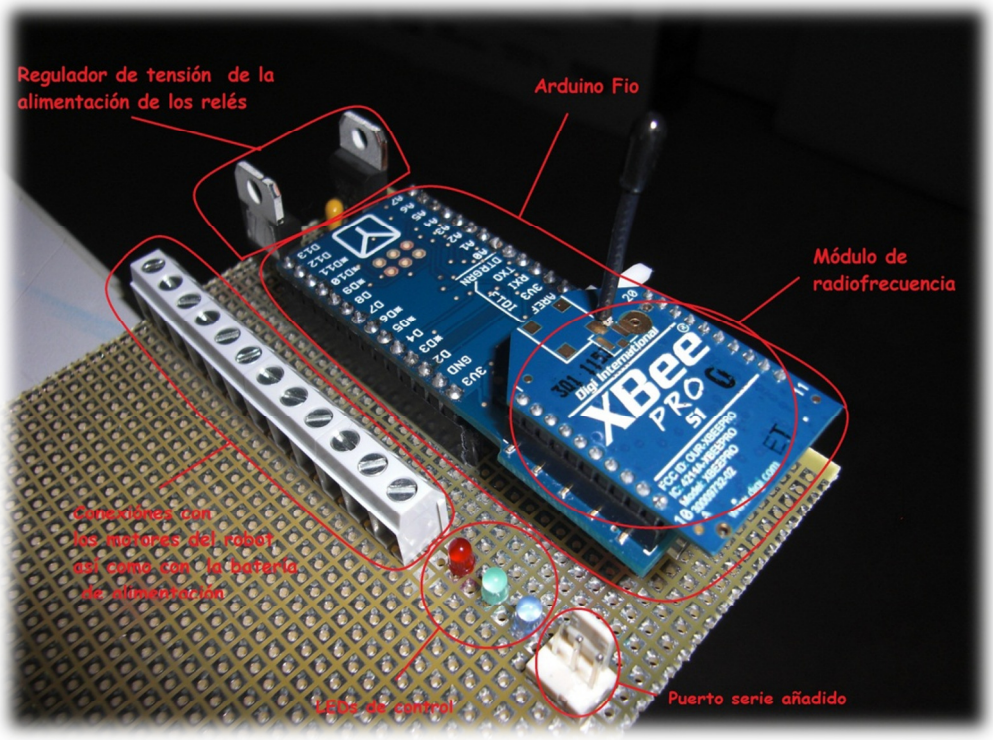


Figura 23: Sistema instalado en el Robot Maggie.

4.2. Desarrollo Software

Una vez definidas las características y esquema de conexión del sistema del robot, se pasa a explicar la programación y el funcionamiento lógico del sistema.

En la Figura 24 se puede visualizar un esquema con los recursos utilizados en este sistema.

En el centro se posiciona el microprocesador y éste actuará sobre ciertas salidas digitales. A su vez también tendrá conexión con el módulo XBee utilizando el conector serial para su configuración inicial y un nuevo *SoftSerial* para poder visualizar comportamientos de las comunicaciones, ya que el serial que trae el microprocesador lo ocupa el módulo de radiofrecuencia Xbee.

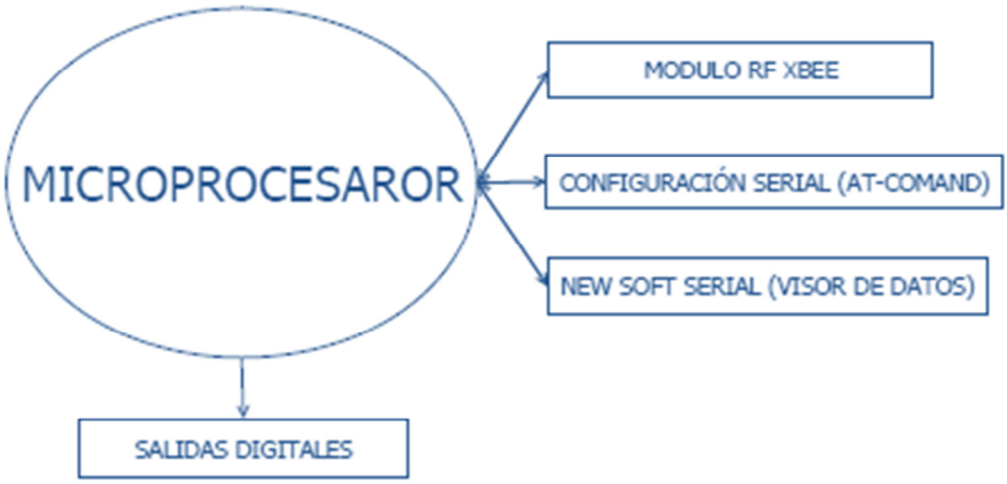


Figura 24: Recursos usados para el sistema alojado en el robot.

A continuación se explica el diagrama de estados general del programa principal (Figura 25), y más tarde se estudiarán otras funciones importantes del programa.

4.2.1. Diagrama de flujo del programa principal

El primer paso que hay que dar es ajustar la configuración de las entradas y salidas, además se debe configurar los parámetros del módulo de radiofrecuencia XBEE. A continuación se inicializan todas las variables que se usarán en el programa.

A continuación enviamos un broadcast (a todas las setas), para que se conecten con el robot.

Una vez enviado el broadcast el robot esperará la respuesta de las setas, por lo que el siguiente paso es leer del Xbee (Figura 26) las respuestas de cada seta, para ello tenemos una función.

A continuación actuaremos sobre los relés que activarán o apagarán al robot. Cabe destacar que si se recibe la señal de emergencia, el programa llama a la función de emergencia, en esta función, además de activar el relé correspondiente, resetea el vector de direcciones y el número de setas conectadas, para que se empiece un ciclo nuevo de nuevas conexiones.

Una vez obtenido el paquete con la dirección de origen guardamos la dirección en un vector de direcciones (Figura 27), donde se quedarán guardadas todas las setas que están asociadas a ese robot, esta función también nos devuelve el número de conexiones que existen en el sistema.

Una vez transcurridos los 300 milisegundos de lectura del Xbee pasamos a hacer una comprobación de si es la primera vez que se

ejecuta el programa, de ser así actualizamos una variable auxiliar con el número de conexiones, esto es así para poder inicializar la

variable auxiliar donde se guarda el número de setas conectadas en el estado anterior.

Si no es la primera vez que se ejecuta el programa, hay que comparar el número de conexiones que había en el bucle anterior con el actual, teniendo en cuenta que se han podido conectar setas nuevas. Si en el estado anterior resulta que había más conexiones que en el nuevo estado menos el número de conexiones nuevas, significará que hemos perdido alguna conexión, lo cual indica una problemática en el sistema y habrá que parar el robot, para ello llamamos a la función emergencia que detendrá el robot.

Si no se ha perdido ninguna conexión actualizaremos la variable auxiliar del número de conexiones anteriores con el número de conexiones actuales.

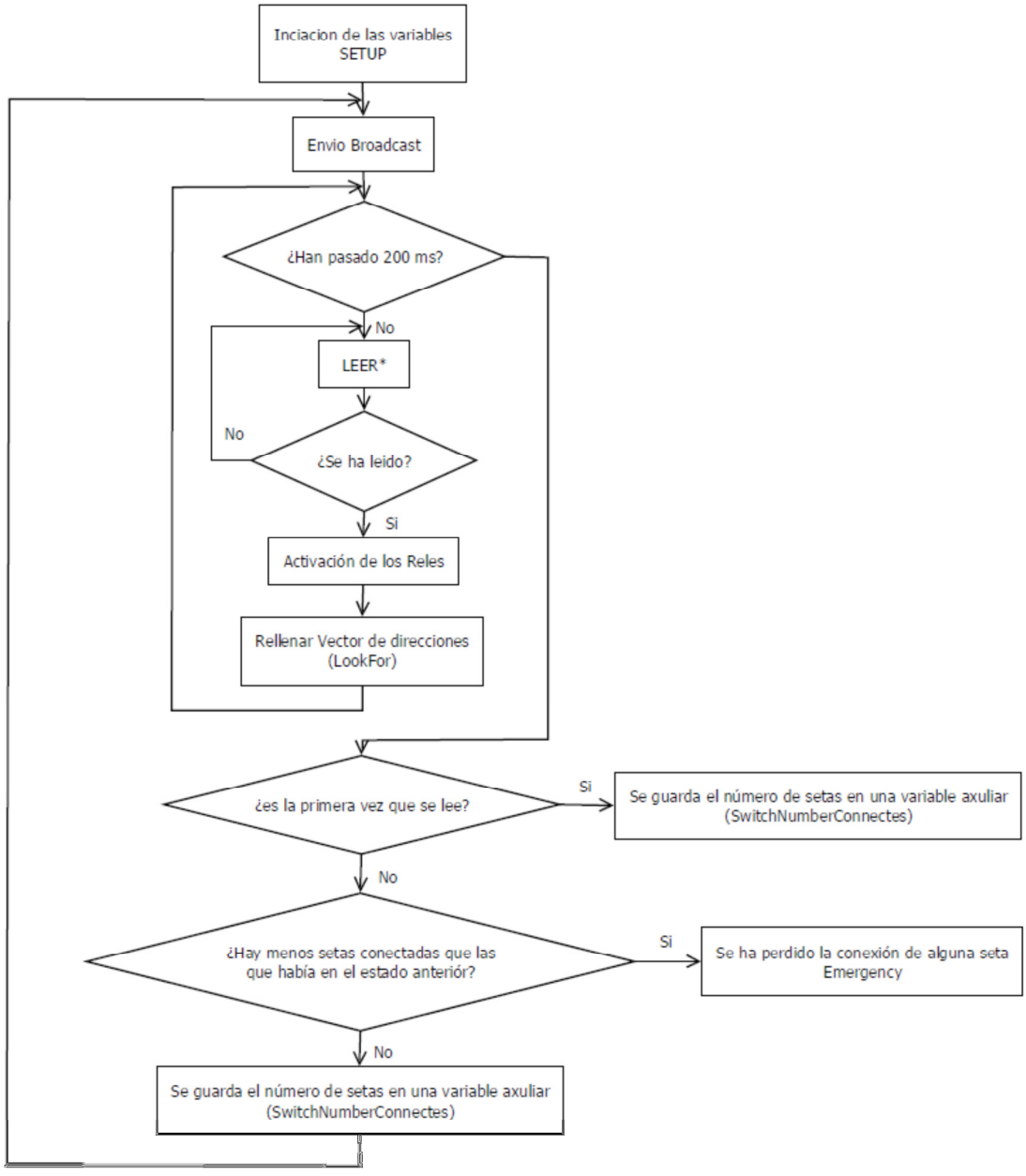


Figura 25: Diagrama de flujo general del sistema alojado en el robot.

4.2.2. Diagrama de flujo de lectura del XBEERead

La función *XBEERead* lee el buffer y obtiene la dirección y el paquete transmitido por el emisor. Esta función devuelve 0 si no se ha leído nada y un 1 si la lectura ha sido exitosa.

Además si se ha leído correctamente el paquete, se tendrá que enviar una señal a la seta para confirmar que el paquete se ha recibido correctamente.

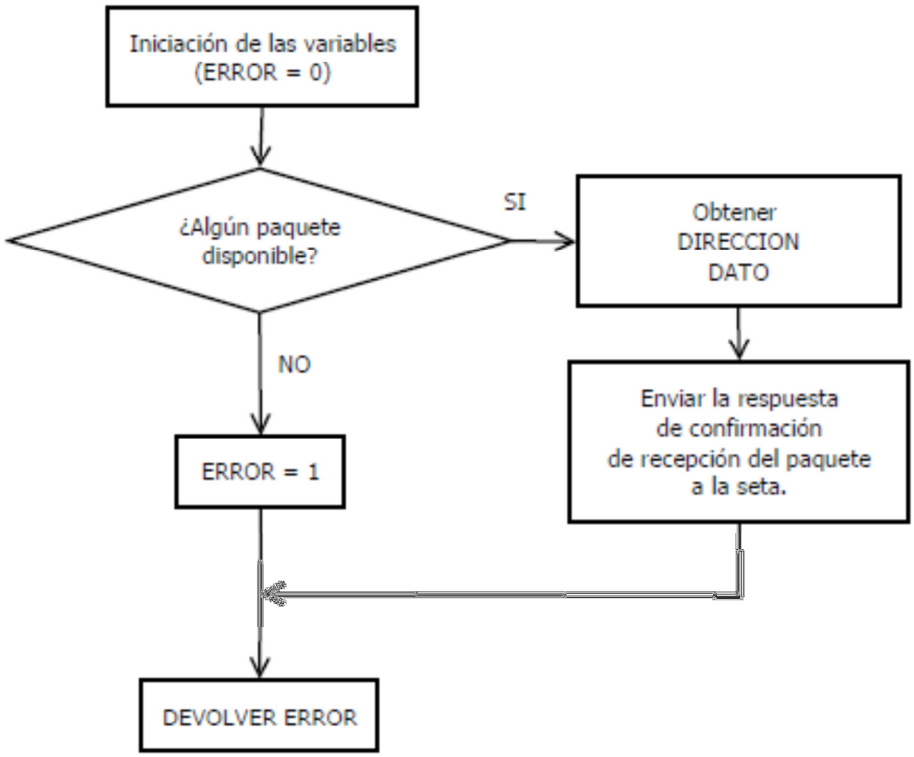


Figura 26: Diagrama de flujo de la función de lectura del Xbee (XBEERread).

4.2.3. Diagrama de flujo de la función LookFor

La función LookFor se usa para buscar e identificar las direcciones de las setas que ya han sido asociadas al robot, y la de registro de la seta en el caso de ser la primera vez que intenta conectarse.

El diseño esta realizado para que cada robot esté conectado con 10 setas a la vez como máximo, por eso justo después de inicializar las variables usadas por la función se deberá preguntar si el número de setas conectadas es menor de 10, si es así se irá recorriendo el vector de direcciones comparando con la dirección actual con la que estemos trabajando.

Hay que comprobar previamente si la dirección actual es la primera dirección, para poder iniciar correctamente el vector de direcciones. Si no es la primera vez se va comparando con todos los elementos del vector.

Si dos direcciones coinciden significará que la seta estaba registrada y se saldrá del bucle. En el caso de que no encuentre ninguna dirección igual, significará que se quiere realizar una nueva conexión, por lo que el vector de direcciones se deberá actualizar.

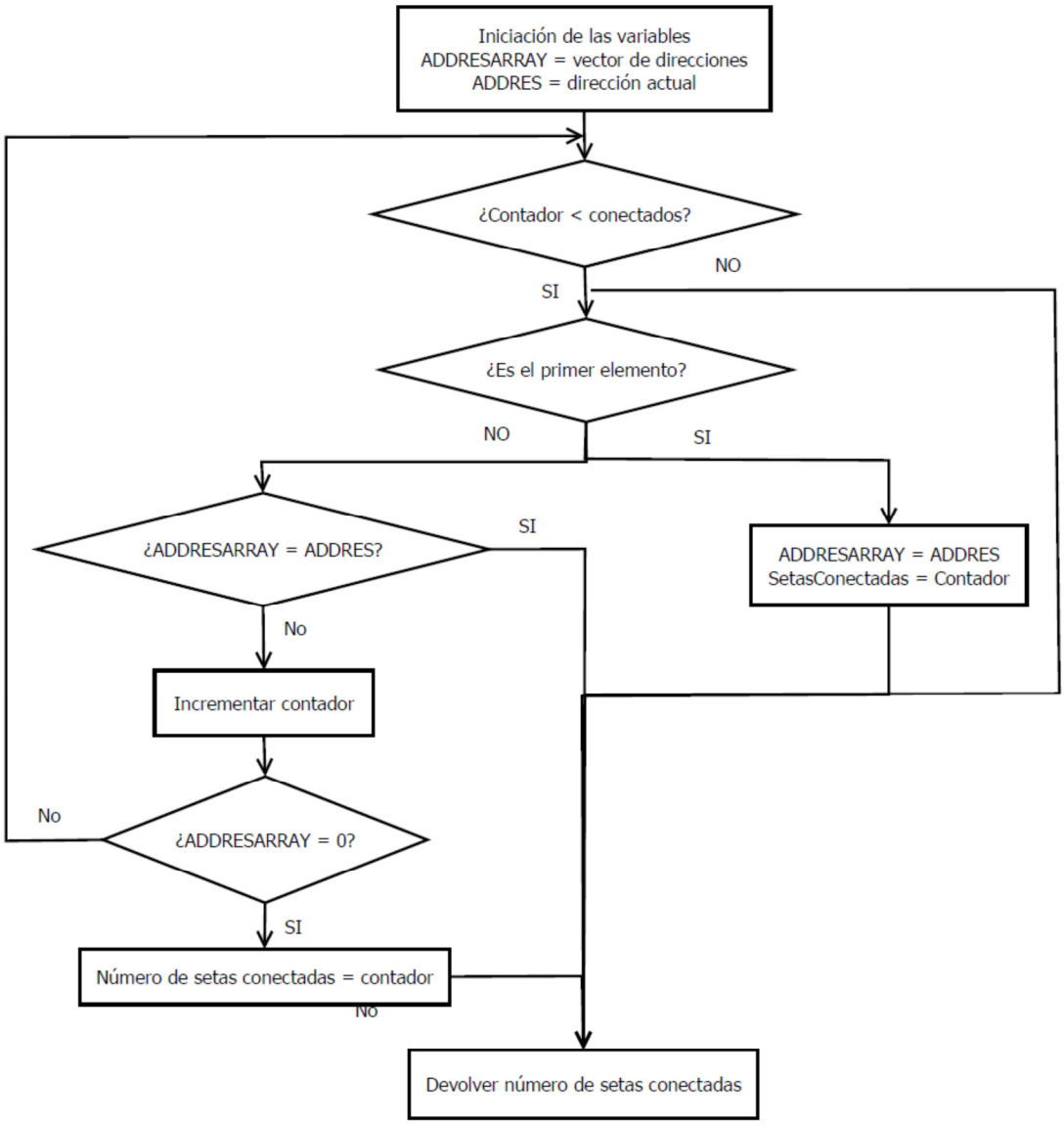


Figura 27: Diagrama de flujo de comprobación del vector de direcciones (LookFor).

5. Sistema embarcado en el mando de la seta de emergencia

La seta de emergencia es el medio por el cual el usuario puede controlar el robot, para poder encenderlo o apagarlo cuando crea conveniente y siempre de la manera más segura posible, ya que podrá realizar esta operación de forma inalámbrica.

Las funciones de este sistema serán entre otras la de respuesta automática cuando algún robot envía la señal de conexión o refresco de setas asociadas y la de lectura de la posición de los pulsadores y envío al sistema alojado en el robot para que éste cambie su estado.

5.1. Desarrollo Hardware

Este dispositivo dispondrá de un microprocesador Arduino Fio junto con un módulo de radiofrecuencia Xbee, al igual que el robot, pero éste podrá elegir a que robot conectarse, asociándose a una u otra red (diferentes PAN ID), pero siempre enviando los paquetes a la misma dirección, tal como se mencionó en el capitulo anterior del sistema alojado en el robot.

El sistema está diseñado para que se pueda conectar con hasta 8 robots diferentes. Esta limitación nos la impone el switch , ya que contiene tres interruptores, por lo que habrá 2^3 posibles redes de asociación, es decir 8 robots diferentes. Además si se pusieran más robots, habría que poner los tiempos de refresco más largos, ya que los tiempos de retardo de envío de los paquetes de cada seta irían aumentando.

Para poder elegir el robot al que se desea conectar, es necesario que ninguno de los sistemas embarcados en los robots estén enviando los paquetes para la conexión de las setas, ya que los paquetes recibidos confundiría a la comunicación serial entre Xbee y Arduino Fio impidiendo una correcta configuración de los módulos.

Como método de seguridad, es necesario que el usuario encargado del control y elección del robot que se desea dominar tenga los conocimientos de lo que conlleva, es decir de manera errónea puedes elegir controlar un robot que no has escogido, pudiendo poner en funcionamiento otro diferente, lo que podría causar un accidente. Para que esto no suceda, se ha colocado el switch dentro de la carcasa, de tal manera que para modificar el robot con el que se conecta la seta, se tenga que abrir la tapa. Además, es necesario cortocircuitar con un

jumper una de las entradas del microprocesador para entrar en el modo de configuración del módulo de radiofrecuencia Xbee. Para usar este pin es necesario configurar una de las entradas analógicas como salida digital, esto es sencillo gracias a la plataforma de programación de Arduino, simplemente hay que definir el pin analógico como pin digital (el elegido el pin 16).

Además, cada seta dispondrá de dos pulsadores para la activación del robot y otro pulsador para la parada de emergencia. También tendrá un sensor táctil con el cuál se podrá visualizar en una barra de 3 LEDs amarillos la carga de la batería, la cuál cuanto más carga más LEDs se iluminarán [12].

Para que el usuario visualice el estado en el que está el sistema y el robot, la seta dispondrá de un LED verde que parpadeará lentamente cuando la seta intente conectarse a un robot, y parpadeará rápido cuando ya se haya asociado al robot, y un LED azul que indicará que el robot está en funcionamiento.

En la Figura 28 se puede ver el esquema de conexiones eléctricas de la seta.

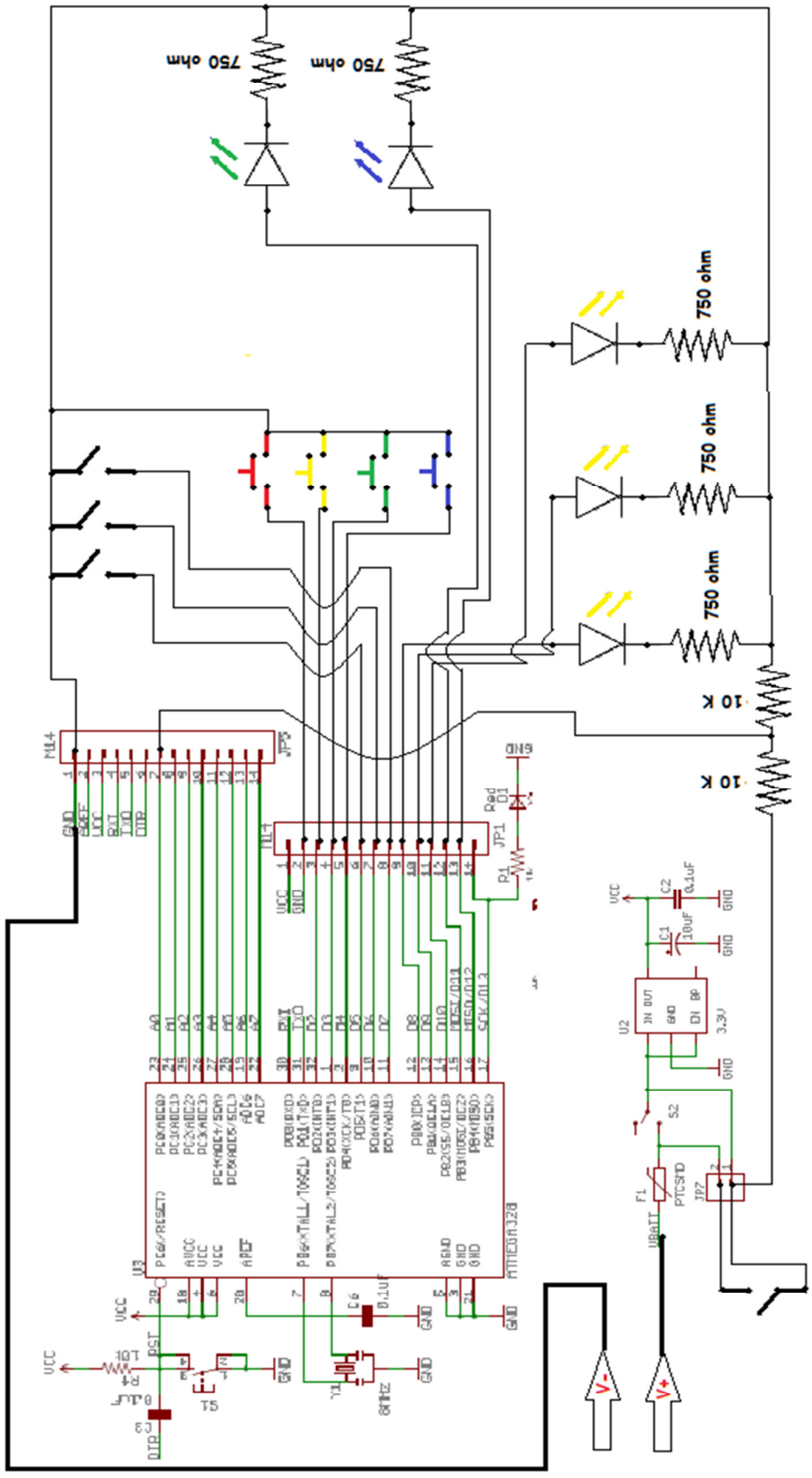


Figura 28: Esquema electrónico de conexiones del sistema de la seta de emergencia.

En la parte de la izquierda de la Figura 28 se puede observar una representación grafica del Arduino Fio. Conectado a las entradas y salidas del microprocesador se conectan los LEDs y los pulsadores tal como se detalla a continuación:

- LEDs amarillos de lectura de tensión de la batería: salidas digitales D11, D12 y D13.
- LED verde de conexión con el robot: salida digital D9.
- LED azul de estado del robot: salida digital D10.
- Switch de elección de robot: entradas digitales D6, D7 y D8.
- Sensor táctil para la lectura de la tensión de la batería: entrada digital D3.
- Pulsadores de encendido del robot: entradas digitales D4 y D5.
- Pulsador de emergencia: entrada digital D3.
- Lectura de la tensión de la batería: entrada analógica A0.

Para poder realizar la lectura de la tensión de la batería ha sido necesario implementar un divisor de tensión previo a la conexión con el microprocesador. Esto es debido a que la máxima tensión que los pines analógicos pueden leer es de 3.3 V. y la tensión de la batería es de 3.7 V. por lo que en el rango entre 3.3 y 3.7 V. no se podrían distinguir valores de tensión.

Para la conexión de los LEDs fue necesario colocar una resistencia en serie con el LED, ya que si no la colocásemos podríamos fundir el LED. Para saber el valor de la resistencia que hay que conectar es necesario resolver la malla de la Figura 29. La ecuación a resolver es la siguiente:

$$V_{PIN} = V_d + I \cdot R$$

Sustituyendo nos queda:

$$3.3 = 1.8 + 0.002 \cdot R \rightarrow R = 750\ ohms.$$

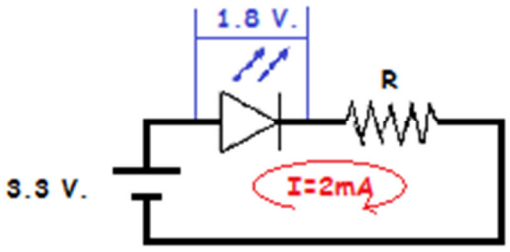


Figura 29: Malla para saber la resistencia que se coloca en serie con el LED.

A la hora de integrar todo el sistema electrónico en su carcasa aparecieron problemas de espacio, ya que se pretende que la seta de emergencia sea lo más pequeña y manejable posible. Pese al pequeño tamaña, se pude introducir todo el sistema tal como aparece en la (Figura 30).

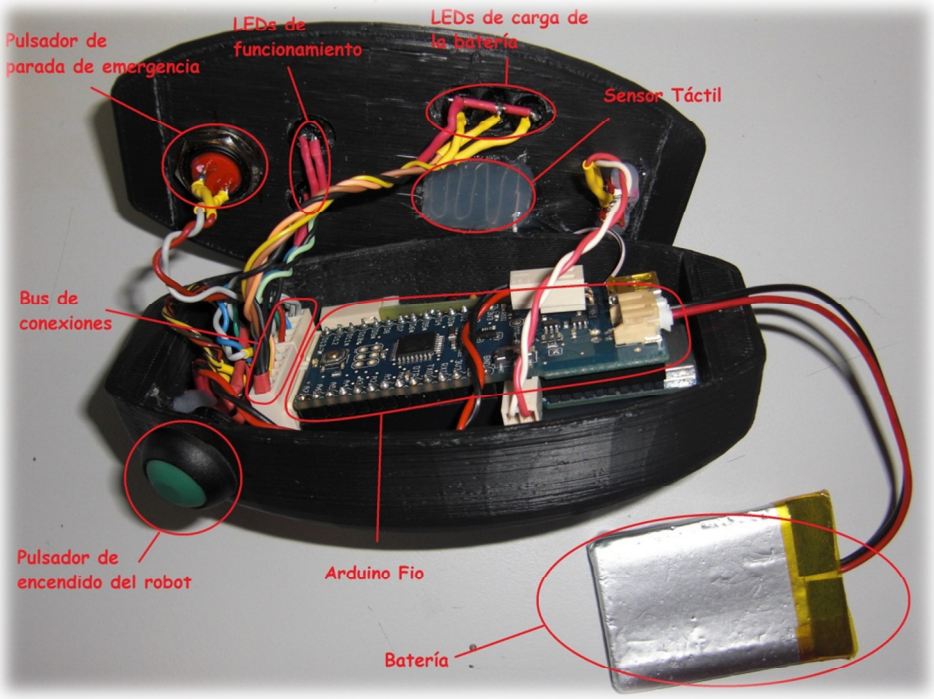


Figura 30: Integración del sistema electrónica en la carcasa.

.

5.2. Desarrollo Software

En la Figura 31 se presenta un diagrama de bloques sobre los principales recursos que utiliza el microprocesador. Estos son prácticamente los mismos que el que utiliza el robot, salvo que el sistema embarcado en la seta de emergencia hace uso de entradas digitales y analógicas además de las salidas que activan los LEDs y también se sirve de la memoria EEPROM del Arduino FIO para escribir y leer dos parámetros importantes para el sistema, uno es la PAN ID a la que se conectará y el otro parámetro es la dirección de la seta. Este último parámetro se usará para calcular el tiempo de retardo de cada seta. Más adelante se explicará cuando se obtienes estos datos y como se debe calcular este tiempo de retardo.

El resto de recursos que utiliza este sistema son relativos a la configuración del módulo de radiofrecuencia Xbee, a la comunicación con el robot y un puerto serie añadido para obtener información relevante en el período de pruebas, que más adelante se desarrollará.

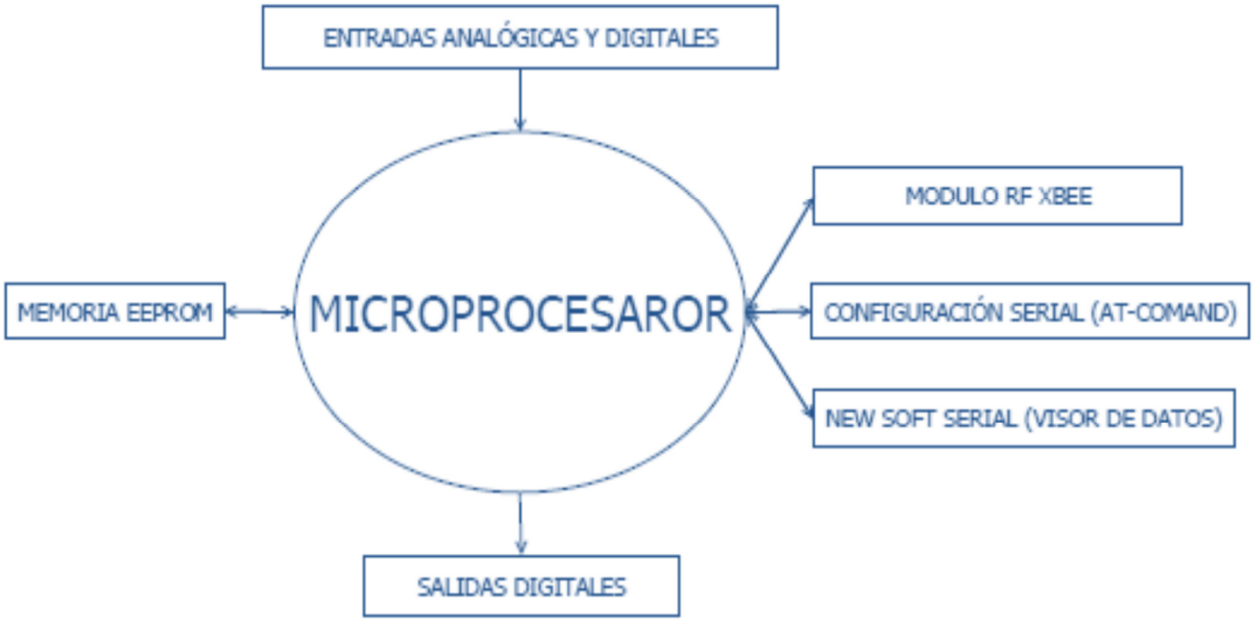


Figura 31: Recursos software utilizados por el sistema alojado en la seta.

El funcionamiento general de la seta consiste en la espera y escucha hasta que llegue algún mensaje del robot, y una vez realizada la recepción del paquete, la seta deberá contestar al robot. El paquete de respuesta será diferente en función de los botones pulsados en la seta, de esta manera habrá 5 tipos de mensajes:

- Ningún botón pulsado.
- Parada de emergencia.
- Botón de arranque 1.
- Pulsador arranque 2.
- Pulsador arranque 1 y 2.

Una vez enviada la respuesta y verificada que la recepción por parte del robot es correcta, tendremos la opción de lectura de tensión si el usuario

quisiera saber la carga de la batería y finalmente se repite el ciclo volviendo al estado de escucha hasta que llegue un nuevo paquete.

5.2.1. Diagrama de flujo general

Al arrancar el programa lo primero que hay que realizar es la configuración del microprocesador y del módulo Xbee [13]. Para ello se especificará la frecuencia de transmisión de datos, que en este caso será de 57600 baudios y también deberemos elegir el robot al que queremos controlar, para ello nos conectaremos a una red o a otra, modificando, en función de los botones de selección de robot de la seta, la PAN ID a la que se conectará.

Asimismo en el setup, se calculará el tiempo de retardo que debe poseer esa seta. Cada seta tiene una dirección diferente y el sistema poseerá hasta 10 setas y las direcciones irán desde la 2 a la 9 y de la A la B con una representación en el sistema numérico hexadecimal. Cada seta tendrá un desfase de 25 milisegundos con respecto a la anterior, por lo que el tiempo de desfase final vendrá definido por la siguiente fórmula:

$$Tiempo\ de\ retardo = (dirección\ de\ la\ seta - 2) \cdot 25$$

Una vez configurado el sistema, se procederá a la inicialización de todas las variables implicadas en el bucle principal.

A continuación pasamos a leer un paquete, si recibe algo se guarda el paquete recibido en la variable estado y si no se ha leído forzamos a que el estado sea 0. Existen tres posibles valores para la variable estado y en función de éstas se realizará uno u otro procedimiento:

- Estado = 0: Hay dos posibilidades, una es que todavía no haya detectado ningún robot, por lo que nos encontraríamos en el estado de búsqueda, por lo que el LED verde parpadearía lentamente (Figura 34), o bien que ya estuviéramos conectados a un robot, pero éste no ha realizado todavía el refresco para ver las setas conectadas, en cuyo caso el LED parpadearía de forma rápida.
- Estado = 1: Significa que se ha detectado un robot, lo cual indica que nos podremos conectar con él y haremos que el LED verde parpadee de formar rápida. Hay que mencionar, que si recibe este estado significará que el robot aún no está en movimiento.
- Estado = 2: Este estado significa que el robot además de estar conectado se encuentra en movimiento, por lo que además de tener el LED verde parpadeando rápidamente, procederemos a encender el LED azul.

El siguiente paso es leer los botones de la seta pulsados y en función de los mismos enviar una respuesta u otra, tal como se explicó anteriormente.

Es en este momento cuando se introducirá el tiempo de retardo, en el que cada seta tendrá retardos diferentes, de esta manera nos aseguraremos que la comunicación entre la seta y el robot sea la correcta. Finalmente se enviará el estado de la seta como contestación al mensaje de conexión o refresco del robot.

A continuación, si se desea saber la carga aproximada de la batería (Figura 33), se procederá a la lectura de la tensión y en función de la tensión de esta encenderemos una cadena de tres LEDs amarillos, de manera que cuanto más carga tenga, mayor número de LEDs se encenderán.

Finalmente se introducirá la función que hace parpadear al LED de color verde (Figura 34) a una velocidad u otra, en función de si la seta esta asociada o no con algún robot. Si la seta de emergencia está en disposición de controlar un robot, este LED tendrá un frecuencia de parpadeo rápida, en cambio si este dispositivo está buscando algún robot con el que conectarse, el LED verde parpadeará de forma más lenta.

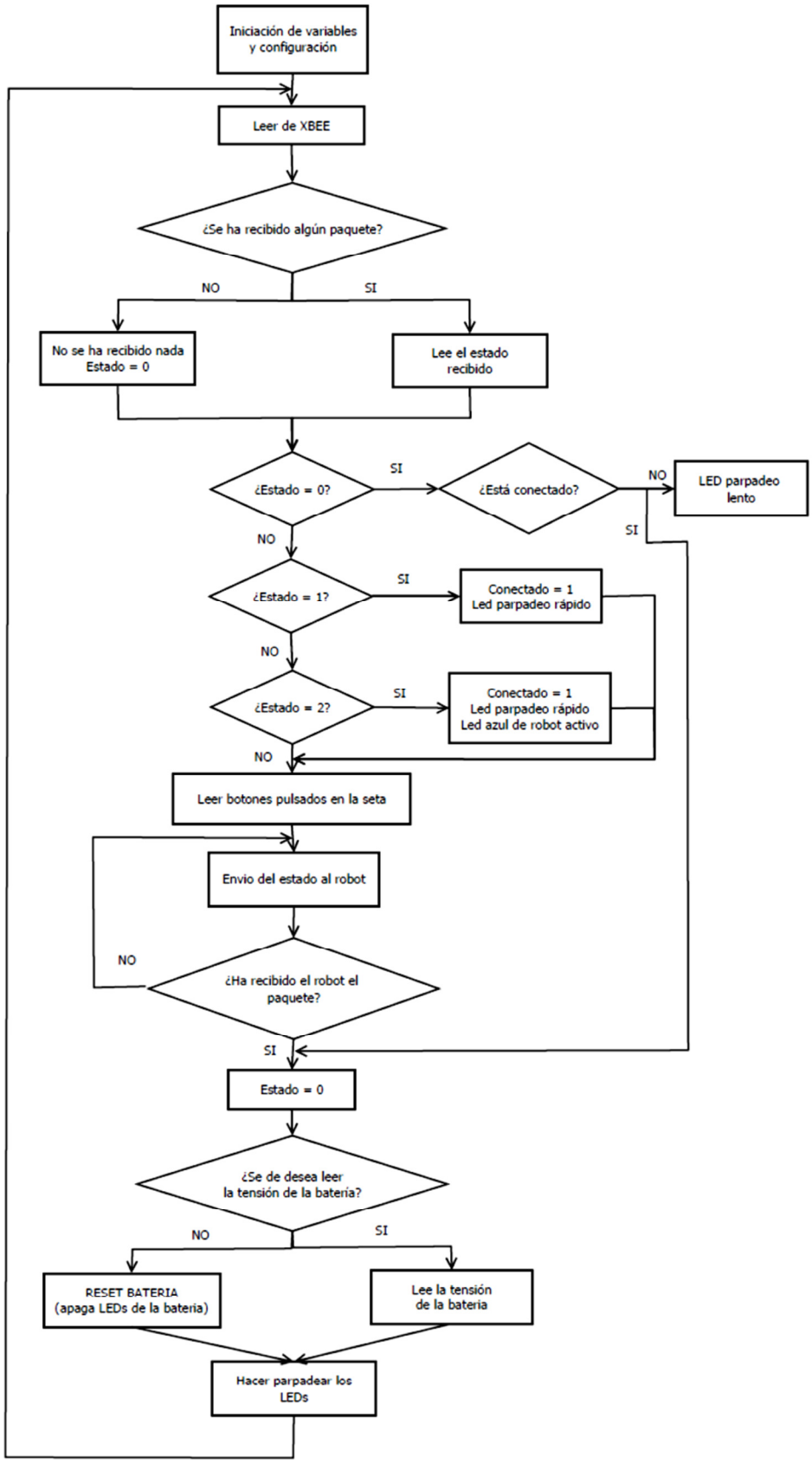


Figura 32: Diagrama de flujo general de la seta.

5.2.2. Diagrama de flujo de la función de lectura de la tensión de la batería

Esta función realiza una lectura (por medio de la entrada analógica del microprocesador) de la tensión de la batería.

Una vez obtenido ese parámetro vamos comparando esa tensión con unos valores fijos de tensión que se estimaron de forma experimental. Más adelante en el capítulo de pruebas se explicará con detalle como se obtienen estos valores. Finalmente en función de estos valores encendemos 1, 2 o 3 LEDs amarillos.

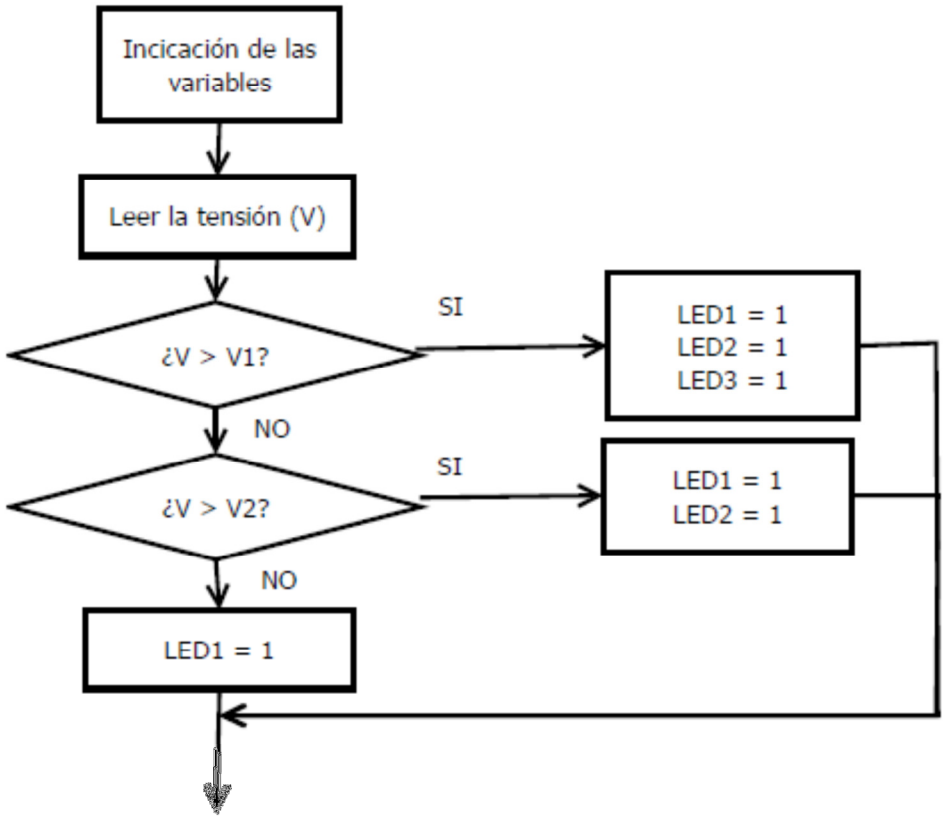


Figura 33: Diagrama de flujo general de la función de lectura de la tensión de la batería.

5.2.3. Diagrama de Flujo de parpadeo del LED verde

Esta función está compuesta por dos partes, ya que para implementarla se realiza utilizando el tiempo del microprocesador como referencia de tiempo.

Para obtener el tiempo inicial del microprocesador para empezar a contar el intervalo deseado, usaremos la función *millis()* que viene predefinida en la plataforma de programación Arduino.

A continuación corre todo el programa principal con sus funciones y al final del *loop* volvemos a medir el tiempo del microprocesador.

Si la diferencia de tiempo entre el tiempo actual del microprocesador y el tiempo obtenido inicialmente es inferior a la frecuencia de parpadeo deseada, no cambiaremos el nivel de salida del pin, en cambio si el tiempo transcurrido entre el inicio y el fin es mayor que la frecuencia, cambiaremos el estado del pin de salida y además se deberá actualizar el tiempo de referencia con el que se compara con el tiempo de microprocesador actual.

Cabe mencionar que esta función de parpadeo del LED se podría haber implementado poniendo delays, pero si los ponemos se descuadrarían los tiempos de retardos y además se tendría al microprocesador parado inútilmente, perdiendo capacidad, incluso podríamos llegar a perder la recepción de los mensajes que envía el sistema del robot.

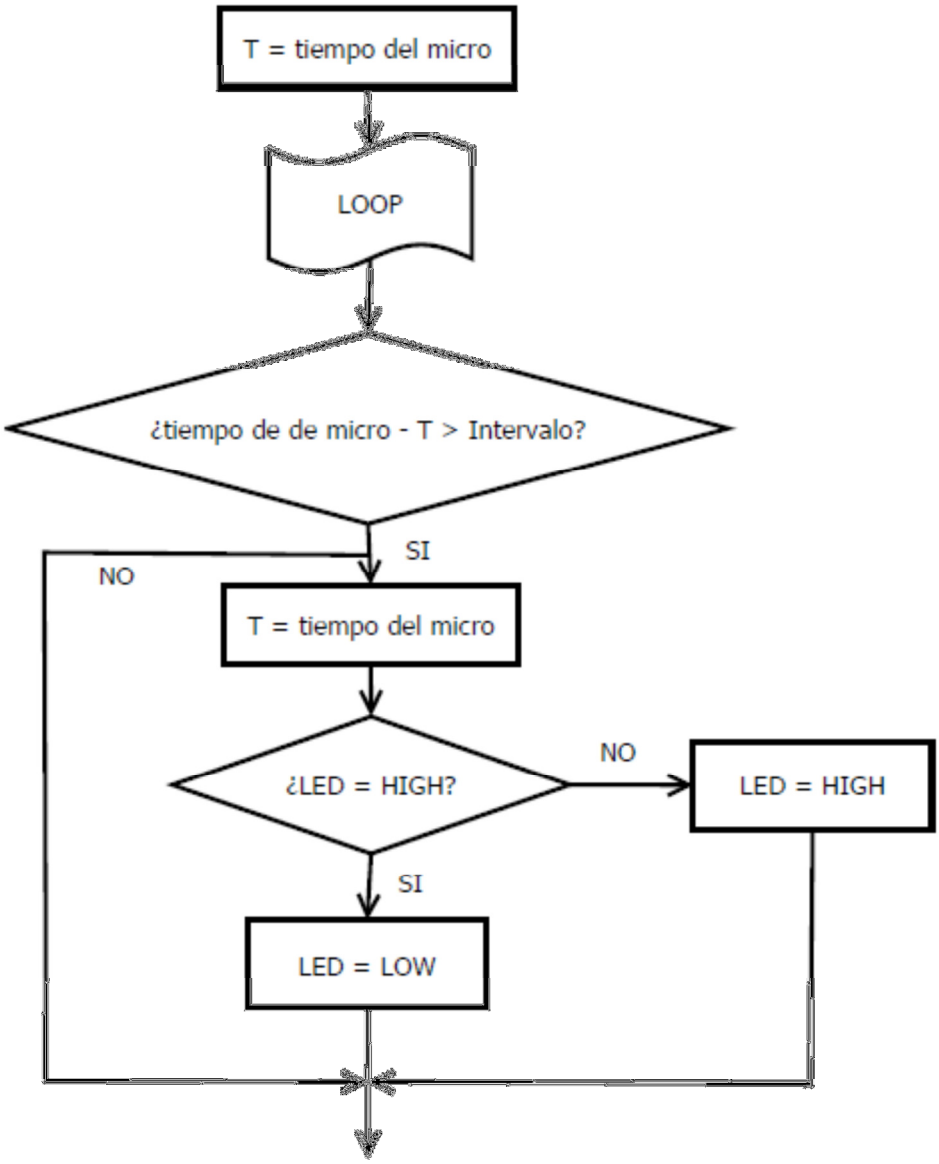


Figura 34: Diagrama de la función de parpadeo del LED verde.

5.2.4. Selección del robot

Para poder elegir el robot al que deseamos conectarnos, todos los sistemas emisores alojados en el robot deben estar desconectados, ya que para poder realizar la configuración de los módulos Xbee para modificar las PAN ID es necesario establecer una conexión serie entre Arduino Fio y Xbee por medio de los ATCommand. Pero el módulo Xbee nunca deja de escuchar de forma inalámbrica, y si hay algún robot pidiendo la conexión de nuestra seta, ésta lo escucharía y tendríamos un conflicto con la información que el microprocesador le comunica para su configuración y lo que el Xbee recibe de forma inalámbrica, por lo que no se podría realizar dicha configuración.

Una vez establecida la configuración deseada se guarda la PAN ID y la dirección de la seta en la memoria interna del Arduino FIO, esta memoria es de tipo EEPROM. Estos datos guardados en memoria se utilizarán más adelante para poder establecer el tiempo de retardo para la contestación de conexión entre robot y seta, tal como se explico al principio del capítulo.

Hay que tener en cuenta que el dato que se recibe cuando se hace una lectura promedio del ATCommand, este ha sido enviado codificado en código ASCII, por lo que para poder usarlo tendremos que descodificarlo pasándolo al sistema decimal. Los números se decodifican restándole 48 al byte, ya que el ‘0’ es el 48, el ‘1’ el 49, el ‘2’ el 50 y así sucesivamente. Pero las direcciones también contienen las letras A, B, D, E y F, ya que están escritas en hexadecimal, pero igual que con los números, estas letras se reciben en código ASCII por lo que al byte de entrada habrá que restarle 55 para pasarlo a decimal.

6. Diseño de la carcasa para la seta de emergencia

Sera necesario diseñar una carcasa para la seta de emergencia, no solo para que el agarre de la misma sea el correcto si no que también esta protegerá al microprocesador y a los circuitos internos ante cualquier impacto.

El material para la construcción de la seta será plástico, ya que es ligero, duro, resistente y de bajo coste.

La técnica utilizada en este proyecto para creación de la carcasa es por medio de la impresora 3D (Figura 35).

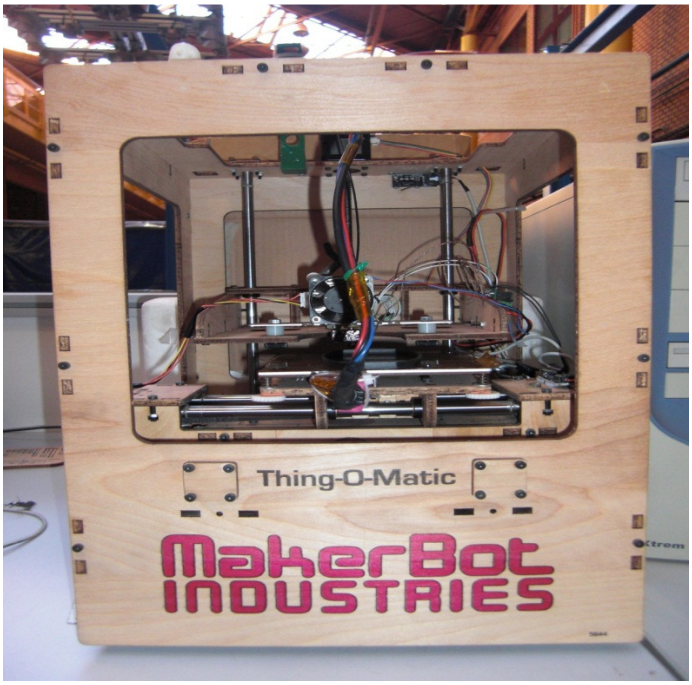


Figura 35: Impresora 3D de MalerBor Insustries.

He utilizado este método porque el departamento de robótica de la Universidad dispone de dos impresoras 3D, por lo que no sería necesario comprar la carcasa y de esta manera se ahorra el coste de diseño y fabricación.

No es el método más rápido, pero para la construcción de pocas piezas es el método indicado. Si quisiéramos hacer muchas más setas, lo que se debería construir es un molde de la seta y por moldeado, soplado o inyección de plástico líquido obtendríamos un mayor número de piezas en menor tiempo.

El funcionamiento es muy sencillo, la impresora consta de dos partes principales:

- La base: es donde se imprimirá la pieza. Esta parte de la impresora posee dos grados de libertad y está situado en el plano horizontal, por lo que se puede desplazar de izquierda a derecha y de adelante a atrás. También, debe estar a una temperatura de unos 120 grados centígrados

para que la pieza se quede pegada y no se mueva durante los movimientos, ya que a esta temperatura el plástico se vuelve pegajoso pero sin llegar a fundirse.

- El extrusor de plástico: este se debe encontrar a unos 220 grados centígrados para poder fundir el hilo de plástico que le llega de una bobina. Además el inyector posee un grado de libertad en el eje vertical, es decir puede desplazarse de arriba a abajo y a la inversa.

Para que la impresora 3D imprima la pieza en cuestión, es necesario que previamente se halla diseñado con algún programa de diseño en 3D. El software utilizado para el diseño de la seta ha sido *Solid Edge*. En la Figura 36 se puede ver el aspecto del *Solid Edge* junto con el aspecto de la base de la seta de emergencia.

La carcasa de la seta esta formada por dos partes, una de las partes es la base de la seta que tiene cierta forma ergonómica y la otra parte es la tapa, que más tarde se atornillará la base.

En la caso de que las piezas no encajen a la perfección, este material de plástico permite realizar, al endurecer, distintos trabajos de mecanizado, como taladrado o fresado.

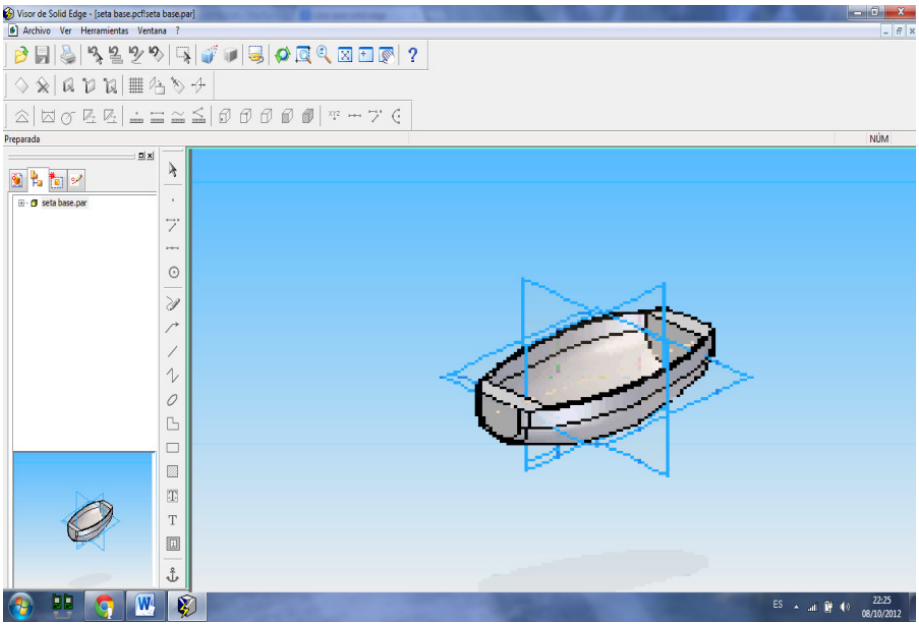


Figura 36: Aspecto del programa Solid-Edge junto con el diseño de la seta.

Una vez realizado el diseño de la pieza en *Solid Edge*, lo importamos a un formato genérico, en nuestro caso el formato elegido es formato STL.

A continuación abrimos el archivo STL con el software que controla la impresora 3D (Figura 37).

Este software nos creará el "gcode", que son las capas necesarias para la impresión de la pieza ya que la impresora 3D va inyectando plástico líquido capa por capa de abajo a arriba y de esta manera se va formando la pieza.

Finalmente, se deberá elegir el espesor de las capas para el contorno y el factor de relleno, ya que al imprimir la pieza, esta no queda totalmente sólida, sino que de forma automática crea una red en forma de panal de abeja. De esta manera se ahorra coste de material y tiempo de impresión, además la estructura de panel de abeja tiene buenas propiedades mecánicas, por lo que la pieza queda igualmente rígida.

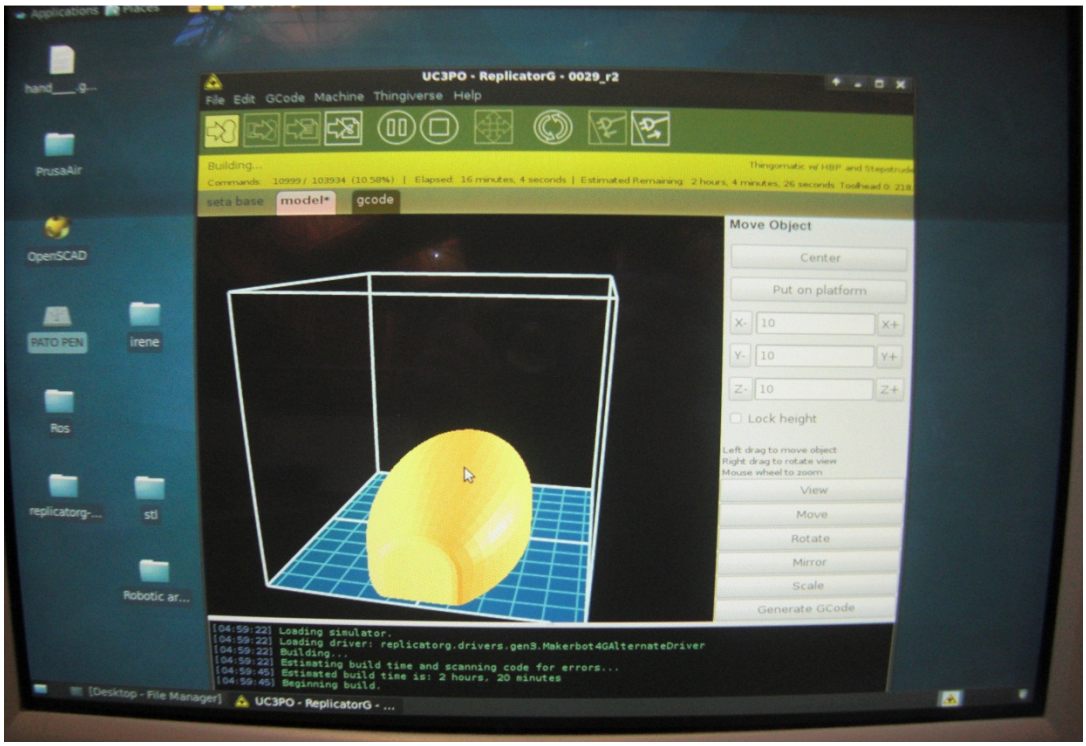


Figura 37: Aspecto del software que usa la impresora 3D.

Este software nos informa de otros parámetros como el tiempo que tarda en construirse la pieza (en el caso de la base fue de 2 horas y 22 minutos), de la cantidad de plástico utilizado, nos da un precio aproximado del coste de la construcción de la pieza, tanto en material utilizado como amortización de la impresora 3D. Asimismo el programa posee un panel de control, donde podemos visualizar la temperatura de la base (debe de rondar los 120 grados para que la pieza quede pegada a la base y esta no se mueva durante la impresión) y la temperatura de inyección de plástico (de aproximadamente unos 220 grados, temperatura a la que se puede inyectar el plástico).

Cabe destacar que a la hora de hacer el diseño hay que tener cuidado, ya que la impresora no puede inyectar tinta en el aire, por lo que hay que jugar con

superficies curvadas para que la siguiente capa de plástico tenga algún punto de sustentación.

A continuación se muestra una secuencia de como se fabricó la base de una de las setas (Figura 38).

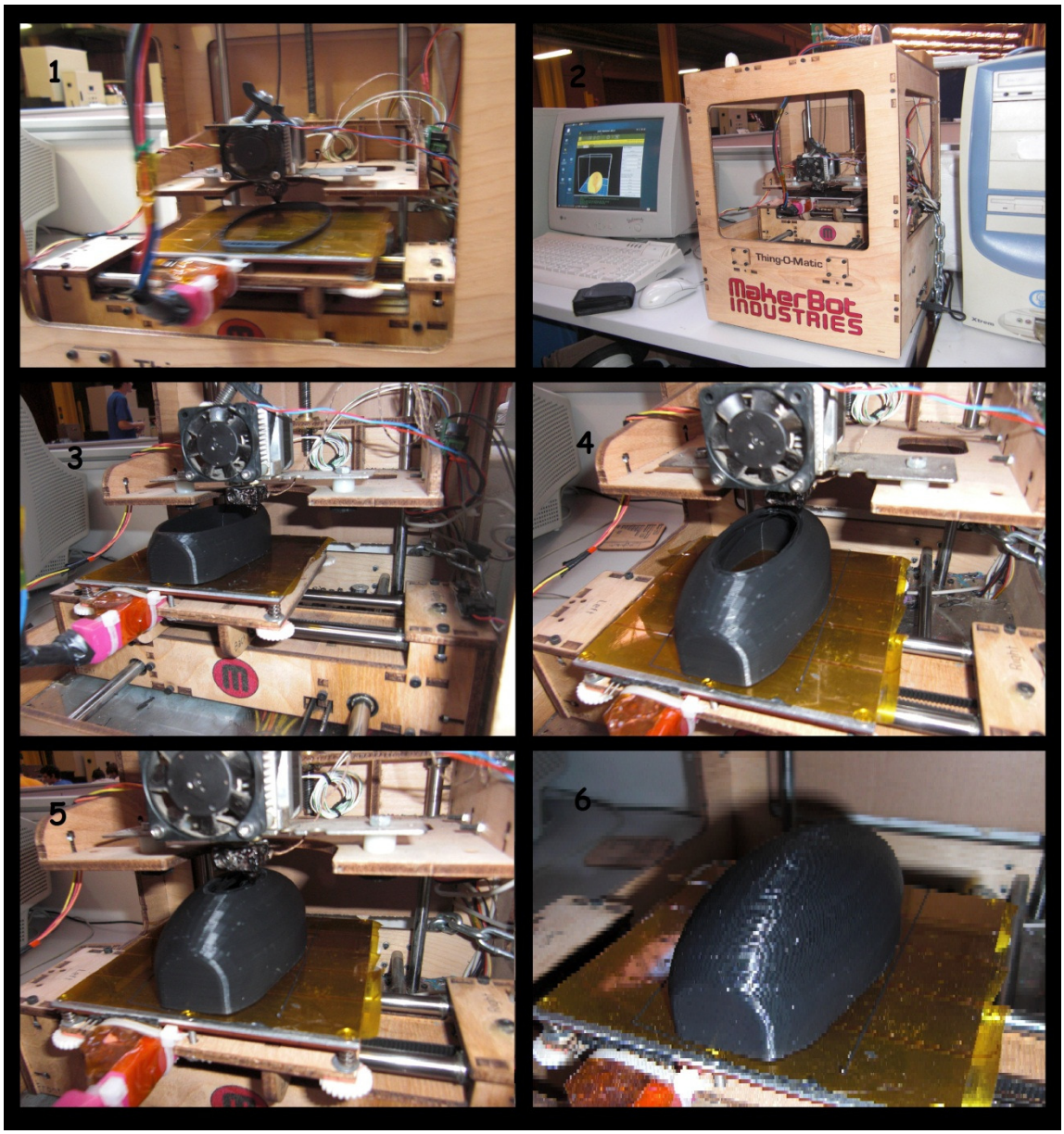


Figura 38: Secuencia de como se va fabricando la carcasa de una seta.

En la (Figura 39) se presenta la carcasa de la seta de emergencia totalmente terminada, en la que ya se han integrado todos los sistemas electrónicos.

Figura 39: Seta de emergencia terminada.

7. Puesta en marcha, pruebas y medidas experimentales

Antes de realizar el montaje final, con la carcasa y colocar todos los componentes de la forma más ergonómica posible, se deben realizar todas las pruebas necesarias para detectar algún tipo de fallo en el diseño y solucionarlo. Para ello se ha montado el microprocesador en una placa, y se han soldado a ella todos los componentes necesarios, tanto para la parte del robot (Figura 40) como para la parte situada en la seta (Figura 41).

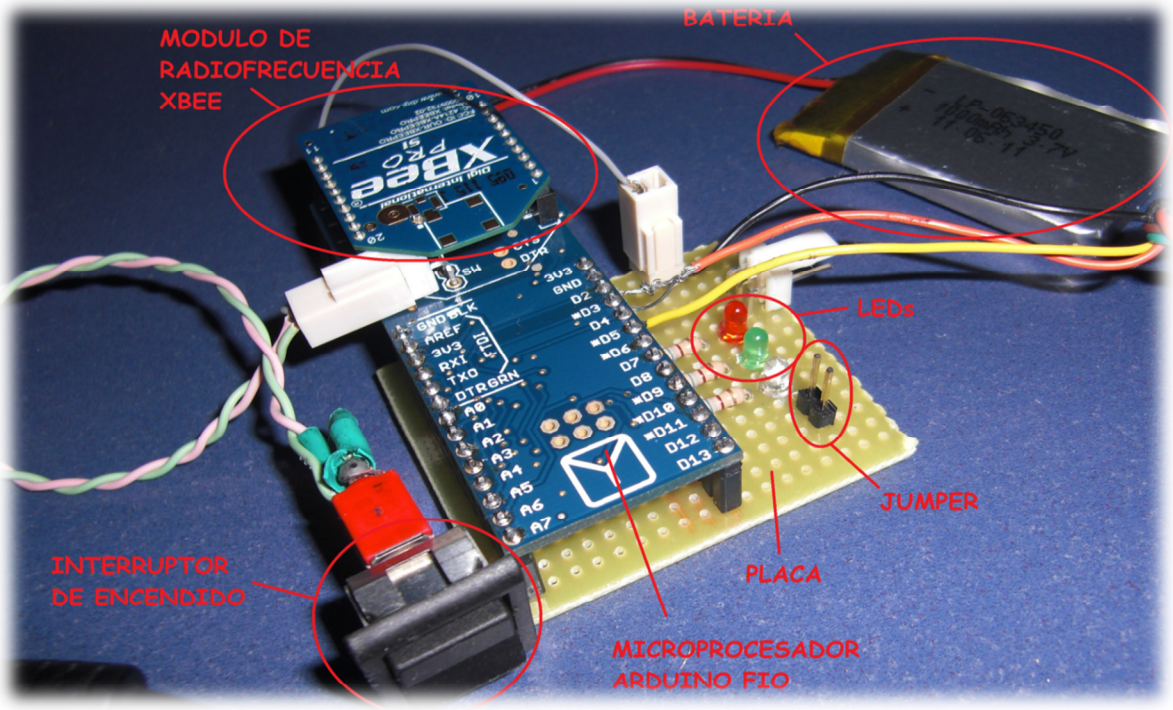


Figura 40: Sistema de prueba alojado en el robot.

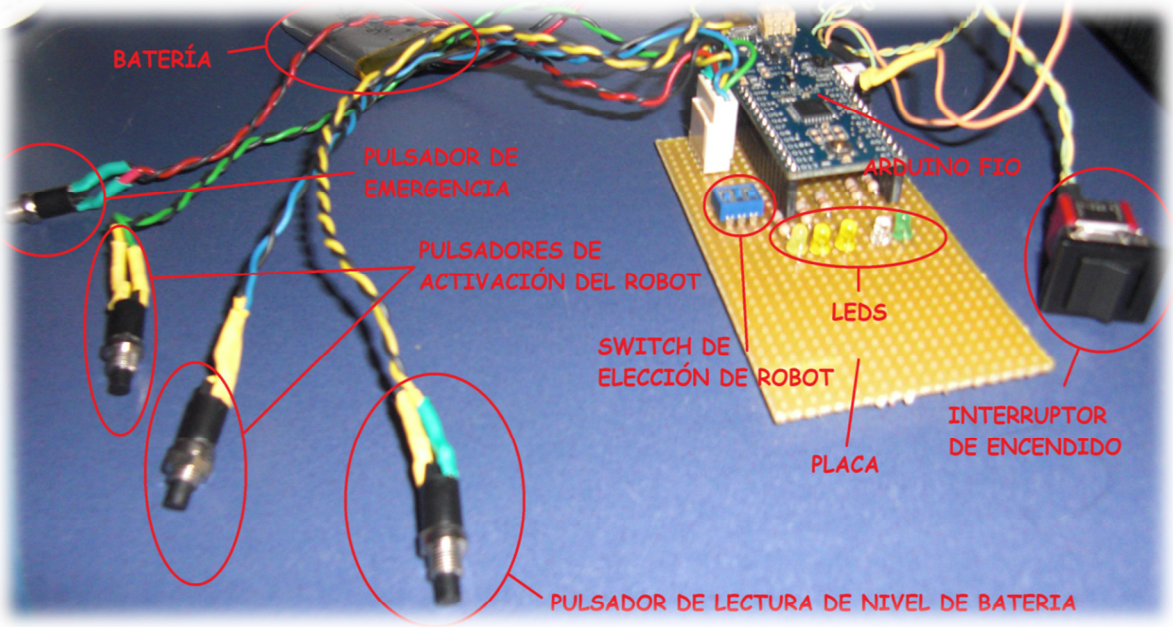


Figura 41: Sistema de prueba de la seta.

Las primeras pruebas que se realizaron, fueron test del hardware, es decir, se comprobó que todos los componentes funcionasen correctamente, tanto los microprocesadores (sus entradas y salidas), LEDs y los interruptores, como todas las conexiones y soldaduras. También se realizó una comprobación de que los módulos XBEE funcionasen correctamente, esto se realizó mediante las funciones ejemplo que proporciona la librería xbee.h [5].

El primer problema que se presentó es que la carga del código del programa en el Arduino se realiza de forma inalámbrica, por lo que cada vez que se carga el Sketch será necesario modificar la configuración del módulo de radiofrecuencia. Esto supone que habrá que sacarlo y meterlo del zócalo repetidamente, lo cual genera fatiga en las patillas del componente, pudiéndose partir en cualquier momento.

Usando un cable FTDI no se solucionaría el problema, ya que para poder conectarlo tendríamos que quitar el módulo también (se podría haber realizado con un cable FTDI y un interruptor, que desconectara todos los pines del módulo, pero es engorroso ya que este módulo dispone de 20 pines, por lo que se desestimó esta solución) [14].

La solución óptima para este problema por la que se optó es la reprogramación automática de los XBEE con la ayuda de "jumpers" (Figura 42). La función del "jumper" es actuar de interruptor, cortocircuitando las dos patillas del mismo. De este modo si el jumper esta conectado, cuando empieza a correr el código y llega a la función del setup, se llamará a una rutina que configurará automáticamente el módulo XBEE para la transmisión de sketches. En el caso de que el "jumper" no esté conectado, el módulo se configurará, en el modo normal de comunicación. Para implementar esta configuración se enviarán las instrucciones por puerto serie (anexo A5) la conexión TX/RX serie para establecer la comunicación usando el modo *AT Command*, al igual que se haría como si se configurara con el XBEE Explorer y un terminal serie del ordenador [15].

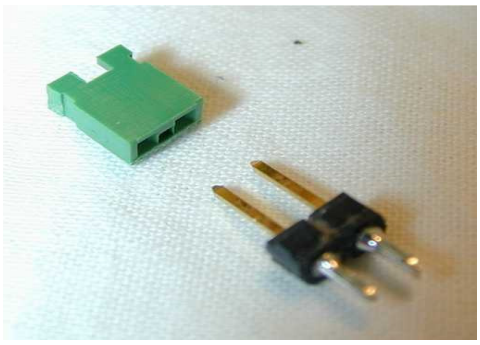


Figura 42: Jumper.

Una vez solucionados todos los posibles problemas e ineficiencias del código se deberán realizar varios test en lo que se refiere a las comunicaciones de tiempo de respuesta.

Para poder obtener las medidas es necesario poder establecer una comunicación serie entre Arduino y ordenador para poder visualizar y guardar en tablas los resultados, para más tarde realizar un análisis de los mismos.

El principal problema es que el puerto serie (TX/RX) de Arduino es usado por el módulo de radiofrecuencia Xbee, por lo que la mejor solución es hacer uso de la librería NewSoftSerial.h [16]. Esta librería permite crear un puerto serie con cualquier salida digital del Arduino Fio. La comunicación se realizará por medio de un cable FTDI, del cual se usarán únicamente 3 hilos:

- TX: para transmisión de los datos.
- RX: para la recepción de los datos.
- GND: para que tanto emisor como receptor tengan la misma referencia de tierra.

Para la implementación hay que realizar una pequeña modificación al hardware y soldar un conector de tres pines a la placa (Figura 43). Los pines elegidos son el 2 y el 3. Se han elegido estos pines ya que el pin GND esta justo al lado, por lo que la construcción es más rápida y eficiente.

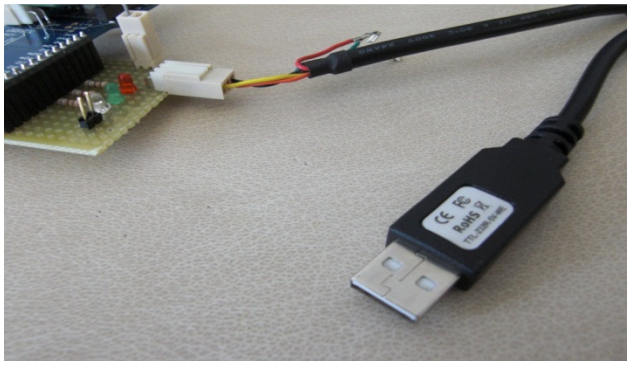


Figura 43: Conexión NewSoftSerial junto con el cable FTDI.

Este montaje se realizó únicamente en el sistema alojado en el robot, ya que es aquí dónde se van a realizar todas las medidas de tiempo.

Las primeras pruebas realizadas son referentes a los tiempos de transmisión y recepción de los paquetes y más tarde se explicarán las pruebas y test realizadas a la batería que está alojada en la seta.

En los siguientes puntos se explicará como se realizaron los ensayos así como los resultados obtenidos.

7.1. Medida del tiempo de confirmación de recepción de un paquete:

En esta prueba se mide el tiempo que transcurre desde que el emisor envía un paquete hasta que el receptor recibe el mensaje y realiza el envío de confirmación de que el paquete se ha recibido correctamente.
Este dato es muy importante ya que es la base para estimar el tiempo que tarda en conectarse una de las setas, así como nos permite establecer el tiempo de espera para que otra seta intente conectarse, es decir cada seta deberá tener un retardo diferente a la hora de enviar el dato (esto se explica en el siguiente apartado).
También nos informa del tiempo que tardaría una seta en enviar una señal de emergencia al robot para su parada. Esta señal debe llegar en el menor tiempo que sea posible ya que es la señal más crítica e importante que la seta puede enviar al robot.

El código de programa utilizado en la seta es el mismo que usa el sistema normalmente.
Al código del robot se le ha realizado una pequeña modificación: antes de enviar el paquete se mide el tiempo del microprocesador (con la función *millis()* de Arduino) y también se mide el tiempo del

microprocesador justo después de comprobar que se ha leído la recepción de un paquete.

Estas medidas se toman cada 2000 milisegundos, es decir, cada 2000 milisegundos se ejecuta la instrucción de envío de un paquete por parte del robot. Se tomaron en total 62 medidas de tiempo. De esta manera obtenemos una nube de datos con los que podremos sacar los estadísticos necesarios y estimar el tiempo que se tarda en la comunicación inalámbrica entre seta y robot

Los resultados obtenidos se presentan en la Tabla 1

Muestra	Tiempo (ms)	Muestra	Tiempo (ms)	Muestra	Tiempo (ms)
1	14	22	14	43	15
2	14	23	14	44	14
3	12	24	14	45	15
4	14	25	15	46	14
5	15	26	14	47	14
6	12	27	13	48	14
7	15	28	14	49	14
8	14	29	15	50	15
9	15	30	14	51	14
10	14	31	14	52	13
11	14	32	15	53	14
12	14	33	14	54	14
13	14	34	15	55	14
14	15	35	14	56	15
15	14	36	15	57	14
16	15	37	14	58	14
17	14	38	14	59	14
18	13	39	14	60	15
19	14	40	12	61	14
20	14	41	14	62	14
21	14	42	14		

Tabla 1: Datos obtenidos de la prueba de medida del tiempo de recepción de un paquete.

Como se puede observar en la Tabla 1 los tiempos de las 62 muestras que se han obtenido oscilan entre los 12 y los 15 milisegundos.

Realizando un posterior análisis estadístico, junto con la representación gráfica (Figura 44) se obtiene que la media de los datos es 14,079 milisegundos, además se ha utilizado otros estadísticos como la mediana y el resultado es 14 milisegundos y la moda que también es 14 milisegundos. Con todo ello se puede concluir y estimar que el tiempo de tardanza en enviar un paquete y recibir la confirmación de recepción del mismo es de 14 milisegundos.

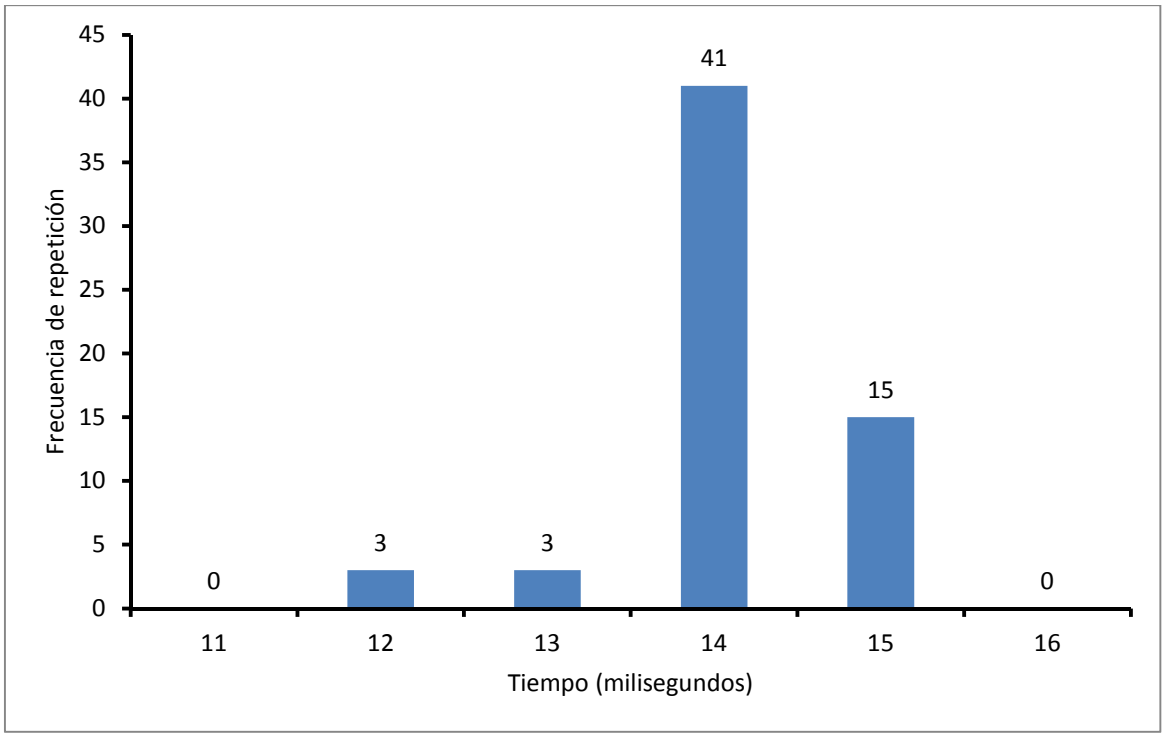


Figura 44: Diagrama de barras de las medidas obtenidas en de la prueba de medida del tiempo de recepción de un paquete.

7.2. Tiempo que tarda en conectarse una de las setas

En esta prueba se pretendía medir el tiempo que tarda en conectarse un número de setas dado cuando las setas envían simultáneamente los paquetes.

Cuando se puso en marcha el sistema se observó que no funcionaba correctamente, ya que los paquetes recibidos por el sistema alojado en el robot solo permiten obtener los paquetes de uno en uno y la librería xbee.h no tiene ninguna función relacionada con la creación una cola con los paquetes recibidos. Esto supone que siempre se perdería la información de alguna de las setas, por lo que el sistema lo interpreta como perdida de conexión y entraría en parada de emergencia.

En un primer intento para solucionar este problema, lo que se implementó fue que cada seta enviase continuamente su paquete hasta que recibiera la contestación de recepción del paquete por parte del robot. Esta solución tampoco fue satisfactoria ya que los tiempos eran demasiado largos, del orden de varios segundos, o bien se producían errores que paraban la comunicación debido a que el sistema alojado en el robot recibía demasiados mensajes y éste se confundía entre todos ellos, a su vez se producían bucles infinitos en la parte de la seta ya que nunca recibía la contestación por parte del robot.

Para poder solucionar este problema completamente cada seta debe enviar su paquete con cierto retardo con respecto a otra seta. Aquí se divide la investigación sobre comunicaciones inalámbricas en dos partes:

- La primera de ellas, es darle un tiempo de retardo arbitrario a cada seta, pero este método no funcionaba correctamente y el sistema alojado en el robot seguía perdiendo la comunicación con alguna de las setas. Esto es debido a que el número aleatorio generado por el microprocesador solo podía oscilar entre el 0 y 275, ya que el robot refresca a todas las setas conectadas cada 300 milisegundos, por lo que es muy fácil que los números generados de manera aleatoria sean tan cercanos que al sistema alojado en el robot no le de tiempo a recibir todos, por esto se desestimo utilizar este método de cálculo de los retardos.
- La forma de respuesta que finalmente se implementó fue darle a cada seta un tiempo de retardo fijo y distinto al resto de setas. De esta manara el envío de paquetes se realiza de forma escalonada, dando tiempo al sistema alojado en el robot a leer todos los paquetes que recibe de todas las setas.

Para saber el tiempo de retardo, nos basamos en el test realizado anteriormente, que nos muestra el tiempo que tarda un paquete en enviarse correctamente. Este tiempo se estimó en 14 milisegundos, pero para que el sistema tuviese en cuenta el tiempo de ejecución de otras instrucciones del propio programa se sumó un margen de seguridad de 1.8 veces el tiempo que tarda en enviarse el paquete, por lo que finalmente el tiempo de retardo que se estableció fue de 25 milisegundos, tiempo más que suficiente para el envío del paquete y

lectura del resto del código, ya que el tiempo que es del orden de microsegundos gracias al reloj del microprocesador de 8 MHz.

Finalmente cabe destacar que el tiempo total de lectura cuando las 10 setas que admite el sistema estén conectadas será aproximadamente de 250 milisegundos, por lo que el tiempo de lectura por parte del sistema alojado en el robot de 300 milisegundos, es más que suficiente para que le de tiempo a recibir los paquetes de todas las setas.

Este tiempo de 300 milisegundos también nos indica el tiempo que tardaría en darse cuenta el sistema alojado en el robot de que una de las setas ha perdido la conexión y entrar en parada de emergencia.

Una vez realzados los test de medida de tiempo en la comunicación se deberán hacer tres ensayos de la alimentación de los sistemas alojados en las setas, ya que estos se alimentarán con pequeñas baterías de 3.7 V.

No será necesario realizar los test para el sistema receptor del robot, ya que este funciona con unas baterías de gran capacidad que utiliza el propio robot como alimentación.

La finalidad de estos test es estimar la carga disponible en la batería mediante la lectura de tensión de la misma.

Esta no es una medida de la carga de la batería exacta, pero existe una relación entre la carga y la tensión que puede proporcionar la batería, es decir, a medida que la carga de la batería disminuye, también lo hará la tensión de la misma.

En los siguientes test se demostrará esta hipótesis y se estimarán los rangos de funcionamiento de la batería cuando esta alimenta al microprocesador Arduino Fio junto con el módulo de radiofrecuencia Xbee, 5 LEDs y dos resistencias del divisor de tensión para poder realizar la medida del voltaje de la batería.

7.3. Medida de la tensión mínima de funcionamiento del microprocesador

En este test se simulará con el uso de una fuente de alimentación (Figura 45) con regulación de tensión conectada a la red. El Arduino Fio será alimentando por esta fuente y se irá disminuyendo la tensión aportada poco a poco hasta que el microprocesador deje de funcionar.

Figura 45: Fuente de alimentación de corriente continúa.

La tensión proporcionada se irá midiendo de dos maneras diferentes: mediante un voltímetro conectado directamente en bornes de la fuente y también se realizará una lectura de tensión mediante una de las entradas analógicas del microprocesador, tal y como lo haría una de las setas para conocer la carga de su batería.
Para poder realizar esta medida, es necesario implementar un divisor resistivo que disminuya la tensión a la mitad, ya que la lectura comenzará en 4.38 V y el microprocesador puede leer de sus pines analógicos una tensión máxima de 3.3 V.

De esta manera, además de obtenerse los limites reales del microprocesador, se obtendrá una nube de puntos que nos determinará si existe algún error entre la medida de tensión real y la lectura del microprocesador por perdidas en las resistencias del divisor resistivo.

Las muestras son tomadas cada 5 mV hasta la tensión de 3.7 voltios y partir de los 3.7 V se tomarán cada 2 mV para tener más datos a partir de la tensión de la batería y así caracterizar de la mejor manera posible el sistema.

En la siguiente tabla se puede observar los datos obtenidos:

Tensión polímetro (V)	Cuentas del micro	Tensión polímetro (V)	Cuentas del micro
4,40	674	3,48	534
4,35	670	3,46	532
4,30	662	3,44	532
4,25	655	3,42	532
4,20	648	3,40	532
4,15	638	3,38	532
4,10	630	3,36	532
4,05	624	3,34	532
4,00	616	3,32	532
3,95	608	3,30	532
3,90	600	3,30	532
3,85	592	3,28	532
3,80	585	3,26	532
3,75	578	3,24	532
3,70	569	3,22	532
3,68	568	3,20	532
3,66	564	3,18	532
3,64	562	3,16	532
3,62	557	3,14	532
3,60	554	3,12	532
3,58	550	3,10	532
3,56	548	3,08	532
3,54	544	3,06	532
3,52	540	3,04	532
3,50	536	3,02	0

Una vez obtenidos los datos pasamos a hacer un análisis de los mismos, para ello se realiza un gráfico (Figura 46) que representa el valor de tensión leída por la entrada analógica del microprocesador, frente a la tensión leída en bornes de la fuente de alimentación.

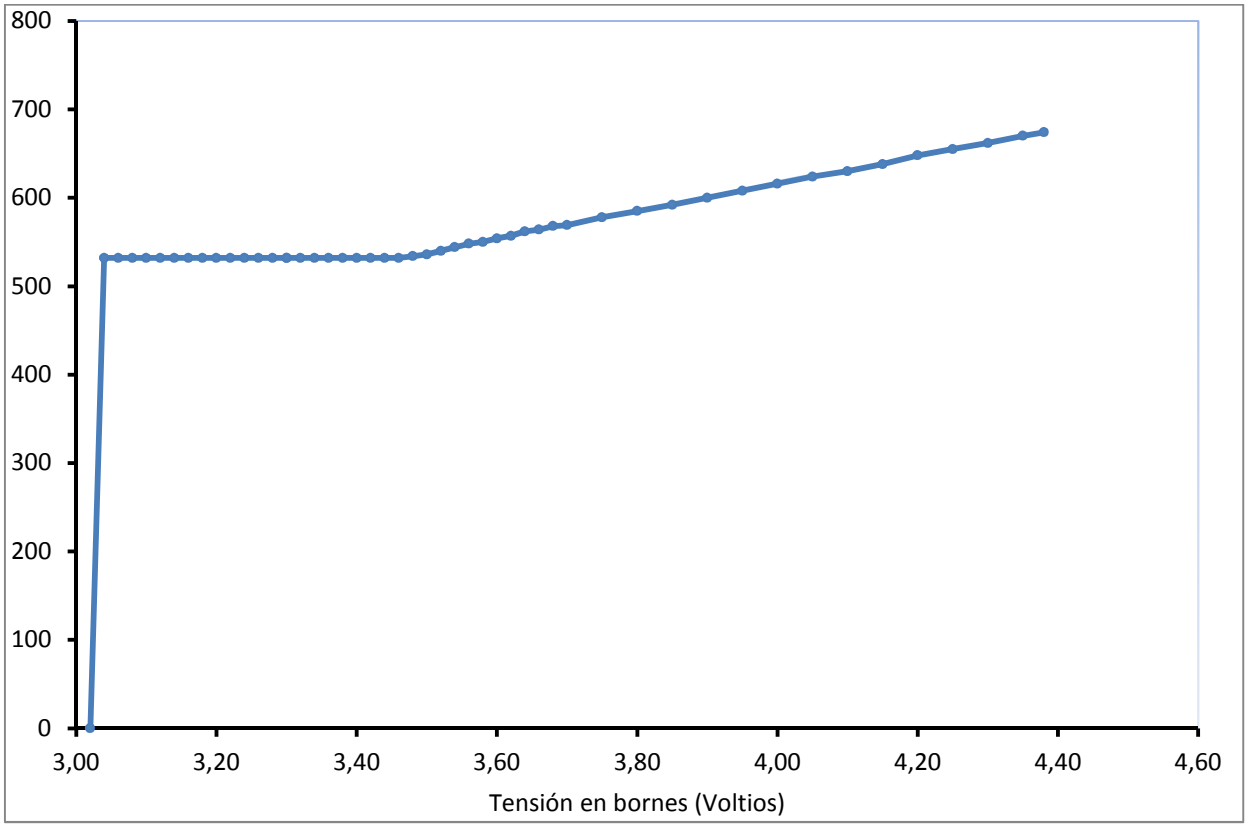


Figura 46: Gráfica que relaciona la tensión de la fuente de alimentación con la lectura del valor de tensión de Arduino.

Se empezó el test con una tensión de 4.38 voltios y se fue bajando la tensión de la fuente de alimentación.

Hasta una tensión de 3.4 voltios se obtiene una relación lineal y la lectura leída por el microprocesador es lineal con la tensión, pero a partir de esta tensión hasta los 3.04 voltios la lectura del microprocesador es constante, pese a continuar bajando la tensión. Esto se podría deberse a que a tensiones próximas e inferiores a 3.3 V el microprocesador no tiene energía suficiente para realizar lecturas correctas por sus entradas analógicas.

Finalmente el microprocesador deja de obtener lectura a partir de 3.02 voltios ya que no tiene suficiente tensión para poder funcionar, y se produce el apagado del mismo.

Realmente el intervalo que nos interesa está entre 3.7 y 3.02 voltios que es el rango de tensiones en las que funcionará la batería de la seta. En este intervalo es donde se produce el punto de inflexión de lectura de tensión del microprocesador, por lo que se podría deducir que entrado en ese punto no distinguiríamos si sigue disminuyendo la tensión de la batería. Por este motivo a la tensión de 3.48 voltios se encuentra un punto crítico en donde se situaría el punto de aviso de poca carga. Para obtener el punto de carga media y alta, hay que observar como se descarga la batería, y ver como evoluciona la tensión de la misma con el tiempo, que es el siguiente test que se realizó a la batería.

Finalmente se hace una valoración sobre que error se comete en la lectura de la batería. Para ello hay que observar que la máxima lectura que puede dar el microprocesador es de 1023, y esto se producirá cuando la tensión sea de 3.3V, por lo que la lectura de la tensión nominal de la batería (3.7 Voltios) traducida en voltios, considerando que se ha disminuido la tensión de lectura a la mitad para que el microprocesador la pueda leer, será de:

$$\frac{1023}{3.3} = \frac{2 \cdot 569}{V} \rightarrow V = 3.67\ voltios$$

Se puede determinar que se comete un pequeño error producido por las pérdidas en el sistema de 0.03 voltios.

7.4. Curva de descarga de la batería

En este test se medirá el tiempo que tarda en descargarse la batería, estando ésta cargada completamente, hasta el punto de que el microprocesador no reciba la suficiente energía para su funcionamiento. Este test se implementará en sistema de prueba alojado en el robot, al igual que el test de medida de la tensión mínima de funcionamiento del microprocesador explicado en el apartado anterior.

Además para que la prueba sea lo más completa posible se realizan dos medidas de descarga de la batería, la primera de ellas es haciendo funcionar al sistema con el mayor consumo posible y otro con el consumo en las condiciones de trabajo.

7.4.1. Ensayo de duración mínima de la batería

El modo de operación se basa en poner el microprocesador en funcionamiento, haciendo que el consumo energético sea suficientemente elevado para que el tiempo de descarga no sea

excesivamente largo y de esta manera saber la duración mínima de la batería.

Para ello se encendieron todos los LEDs que dispone el sistema, tres LEDs del montaje y un LED que posee el propio Arduino Fio alojado en el pin 13. A su vez, para que el consumo fuera algo mayor se puso a trabajar al módulo Xbee enviando señales continuamente.

La prueba consiste en tomar muestras del valor de la lectura de la tensión obtenida a través del pin analógico A0 y del tiempo del micro desde el arranque del mismo con la función *millis()* cada 30 milisegundos aproximadamente, estas muestras se envían a través del montaje del puerto serie de los pines 2 y 3, para más tarde poder analizar todos estos datos.

Todos los datos obtenidos se adjuntan en lo anexo de 4 donde se puede observar todas las muestras de tiempo y lectura de tensión (Tabla 7).

Lo que realmente interesa de este test es observar la distribución de esos datos. Para ello se realiza la representación grafica de la tensión leída frente al tiempo transcurrido desde su arranque (Figura 47).

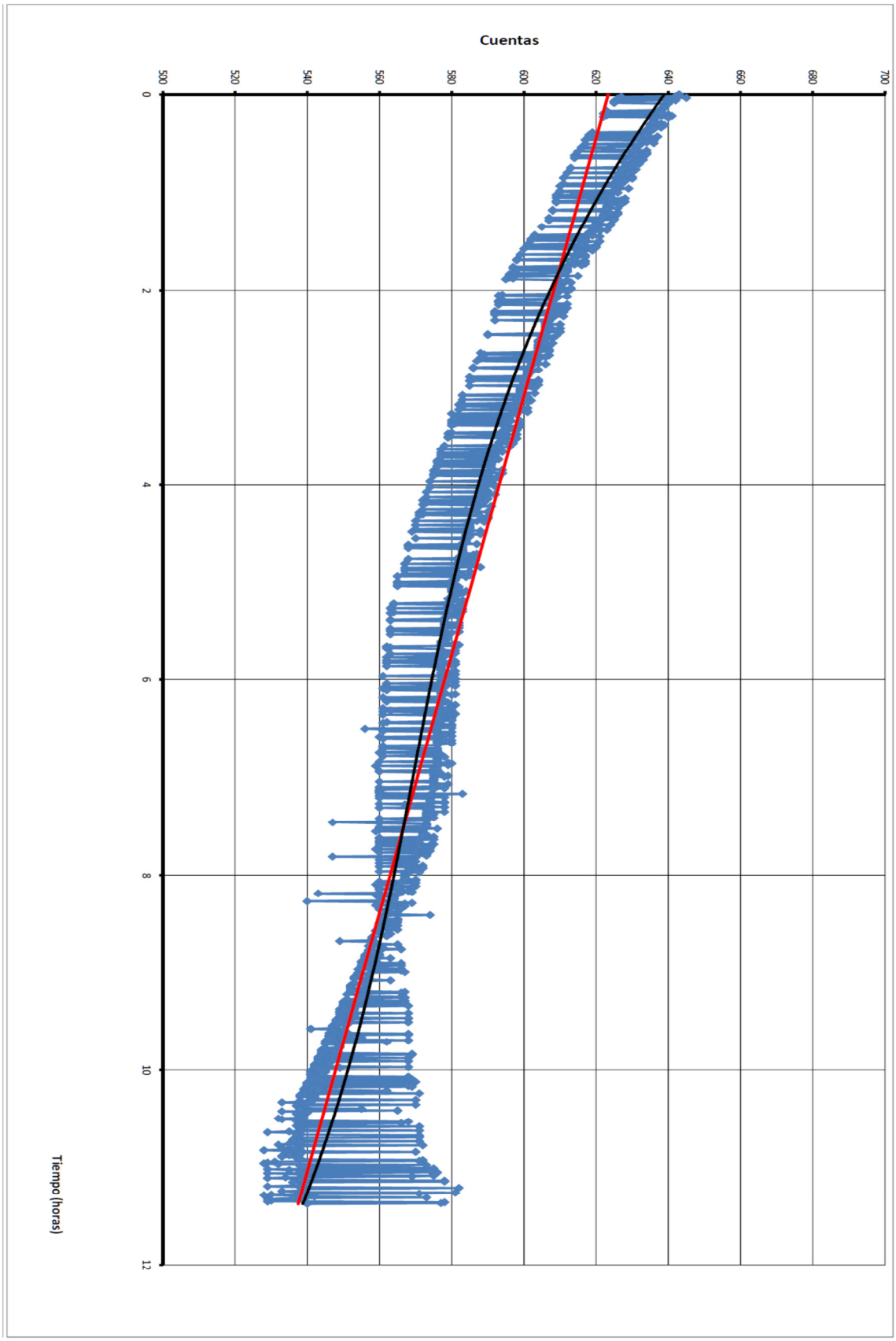


Figura 47: Curva de descarga de la batería (mínimo tiempo de duración).

Lo primero que se observa en la Figura 47 es que se distinguen dos grupos de puntos, esto es debido a que cuando el consumo energético es mayor, la tensión de la batería disminuye, y en este sistema obtenemos puntos de mayor consumo (puntos con menor lectura de tensión) cuando enviamos señales por el módulo de radiofrecuencia, ya que es el componente con mayor demanda energética.
A continuación se realizan dos ajustes lineales a esos puntos:

- En un primer ajuste (curva de color negro), se ha utilizado un ajuste polinómico de grado tres. Como se puede observar este gráfico es más fiel a los datos correspondientes a menor consumo energético, y los muestras de mayor consumo energético desplaza la curva hacia abajo. La ecuación de la curva obtenida es la siguiente:

$$V = -2x10^{-21}\, t^3 + 2x10^{-13} t^2 - 5x10^{-6} t + 623{,}37$$

- El otro ajuste realizado es un ajuste lineal (línea roja de la gráfica). Este ajuste es mucho más sencillo que el anterior y no representa tan fielmente todos los puntos, a pesar de ello, no se aleja mucho de esos puntos, sobre todo cuando nos movemos por puntos intermedios de tiempo. Se produce mayor error en los extremos de la recta, donde se observa que los puntos reales sí se alejan más de la recta de ajuste. En la siguiente expresión se observa esta recta de ajuste:

$$V = -2x10^{-6}\, t + 623{,}37$$

A continuación vemos el tiempo máximo que ha durado la batería, observando la Tabla 7 del anexo, el último dato obtenido es de

40939026 milisegundos lo que equivale aproximadamente a 11 horas y 22 minutos.

7.4.2. Ensayo de duración de la batería en condiciones normales de operación

Para realizar este test se cargó el código de la seta y se puso en funcionamiento junto con el sistema del robot, por lo que el sistema funciona como normalmente lo haría, es decir el robot enviaría el broadcast para detección de setas, y la seta respondería.

La única modificación realizada es que se mediría el tiempo del microprocesador y la tensión de la batería por medio de la entrada analógica A0 cada minuto aproximadamente.

Se ha aumentado el intervalo de tiempo con respecto a la prueba anterior para que el número de muestras obtenidas no fueran excesivamente numerosas, ya que se estima que la duración de la batería sea superior en este caso.

En el anexo 5 se pueden observar todos los datos obtenidos, pero como en el test anterior lo que realmente interesa es la relación que tiene la lectura de tensión y el tiempo transcurrido, que se puede determinar visualizando la gráfica que se muestra a en la Figura 48.

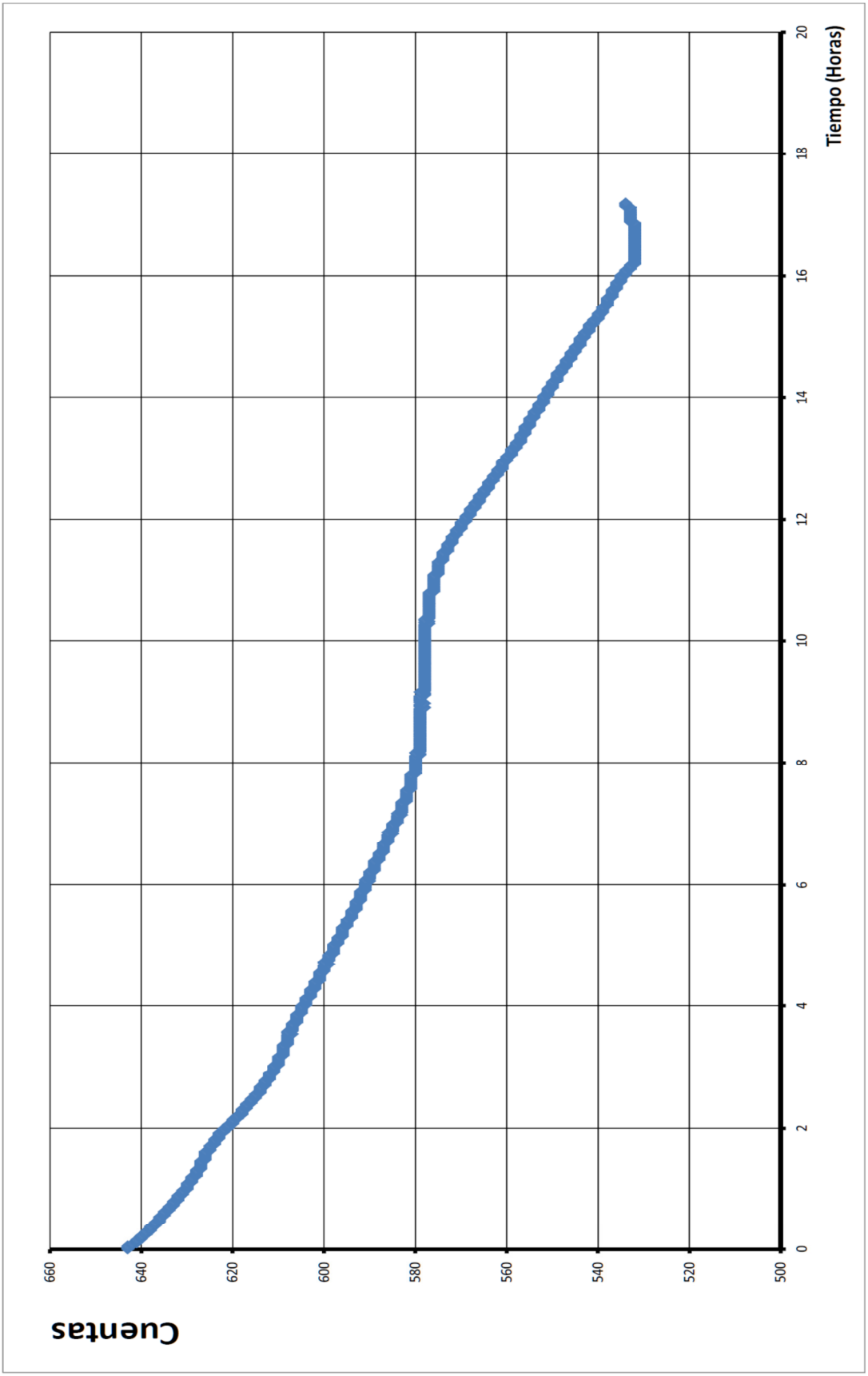


Figura 48: Curva de descarga de la batería (condiciones normales de operación).

Si visualizamos los datos obtenidos (Tabla 8 del anexo) obtenemos una duración de la batería cuando el sistema funciona en condiciones normales de operación de 61882814 milisegundos, o lo que es lo mismo 17 horas y 11 minutos.
Esta cifra es notablemente superior a la del caso de máximo consumo de una duración de 11 horas y 22 minutos. Esto quiere decir que la hipótesis de tomar las muestras cada 60 segundos en vez de cada 30 segundos es correcta.

Una vez analizada la duración total de la batería, observamos un comportamiento diferente en la curva en comparación con el anterior test. Ahora tenemos una única nube de puntos alineada y no dos, esto es debido a que el consumo energético no tiene tantas alteraciones y además éste es más reducido que en el caso anterior, por todo ello no es necesario realizar ningún tipo de ajuste polinómico y mucho menos lineal.

En la gráfica se puede observar también como la tensión va cayendo conforme avanzamos en el tiempo, hasta transcurridas unas 8 horas donde nos encontramos un punto de silla.
En este punto se estabiliza más la tensión y permanece casi constante, aunque tiene una leve inclinación negativa durante las siguientes 2 horas de funcionamiento.

En las siguientes 6 horas, la tensión vuelva a caer al mismo ritmo que lo hacía en el instante inicial de la prueba y cuando llega a la hora 16 desde que se inicio el test vuelve a estabilizarse durante unas dos horas.
Finalmente la batería no tiene suficiente energía para alimentar al sistema transcurridas 17 horas desde el inicio de la prueba.

Estas dos pruebas además de determinar los limites de funcionamiento y la duración de la batería del sistema alojado en la seta, es fundamental para poder estimar la carga de la batería en tiempo real cuando el usuario lo solicite.

La seta dispondrá de tres LEDs, por lo que obtenemos 3 intervalos posibles para el encendido de los LEDs:

- Lectura > 593 (correspondiente a las 5 horas y 43 minutos de funcionamiento): Todos los LEDs encendidos (carga máxima).
- 593 > Lectura > 574 (correspondiente a 5 horas y 43 minutos y las 11 horas y 19 minutos de funcionamiento): Dos LEDs encendidos (carga media).
- Lectura < 569 (a partir de las 11 horas y 19 minutos de funcionamiento): Un único LED encendido (carga mínima).

Como se puede comprobar se deja un gran margen de seguridad (de unas 6 horas de funcionamiento) al usuario para advertirle que la batería esta próxima a agotarse.

8. Conclusiones y trabajo futuro

8.1. Conclusiones

Como conclusiones finales de este trabajo podemos establecer:

- Con el nuevo sistema de arranque y parada de emergencia se ha conseguido que el robot trabaje con mayor seguridad ya que podemos poseer el control del mismo en todo momento. Incluso si una de las setas falla, el sistema implementado lo detectaría y pararía al robot.
- Además de esta modificación de la forma de operar con respecto al sistema anterior, el nuevo proyecto permite operar desde distancias más lejanas, unos 25 metros, ya que el radio de

operación del módulo XBEE SERIE 1 en su versión PRO permite manejar distancias mucho mayores que las que nos encontramos ya que la tecnología utilizada para este proyecto es mucho más potente que la que usa el antiguo sistema de radiofrecuencia convencional.

- Otra de las mejoras con respecto al sistema anterior es que podemos controlar los robots independientemente, es decir es el usuario quien elige que robot desea controlar.

Todo ello se ha podido realizar gracias a la exhaustiva investigación sobre microprocesadores y redes inalámbricas para la transmisión de datos y concretamente estudiando en profundidad al Arduino Fio y los módulos de radiofrecuencia de Xbee. A su vez, he podido adquirir más conocimiento sobre como se deben realizar los ensayos, como por ejemplo los test de la batería, ya que se debe estudiar múltiples casos posibles simulando varias situaciones de funcionamiento, para poder caracterizar de la mejor manera el comportamiento del sistema en su conjunto.

8.2. Trabajo futuro

Como se ha explicado a lo largo de todo este documento, la herramienta fundamental para la implementación del sistema es el uso de un microprocesador. Este es un componente muy potente, con el que se puede realizar muchas más operaciones y se podría haber hecho más uso de su capacidad del que se le ha dado, es decir nos sobran recursos

y capacidad de procesamiento. Esto quiere decir que se podría ampliar la comunicación con más información por parte del robot, lo cual nos permitiría conocer más parámetros del robot, como por ejemplo, la carga de su batería, que funciones está ejecutando en ese mismo momento, datos relativos de los sensores que dispone. Todo ello se podría enviar por los módulos del Xbee y sería recibido por la seta.

En las setas también se podría colocar una pantalla para poder visualizar todos los datos obtenidos del robot, ya que se tiene pines del microprocesador desocupados. Pero esto implicaría mayor consumo energético, por lo que también hay que llegar a cierto compromiso entre funcionalidad, consumos y la finalidad de la aplicación.

Finalmente, se podrían hacer mejoras en el modo de funcionamiento en la comunicación. Para ello habría que hacer un análisis de todas las librerías que nos permiten controlar los módulos de radiofrecuencia, para poder solucionar algunos detalles, como por ejemplo los tiempos de retardo en la comunicación, si se encontrara la manera de formar una cola de todos los mensajes recibidos, no sería necesario los retardos, pero la librería utilizada no nos permitía crear dichas colas, por lo que desfasar el envío de cada paquete cierto intervalo de tiempo es absolutamente necesario. Esta mejora implicaría menor tiempo de comprobación de setas conectadas, lo que haría al sistema más seguro. También permitiría que al robot se le pudieran asociar muchas más setas, por lo que haríamos un sistema más potente y de mayor calidad.

Todo lo explicado anteriormente son posibles mejoras, pero siempre hay que tener en cuenta las especificaciones de trabajo requeridas para este sistema. Se parte de la base de que lo que se pretendía realizar es la mejora del sistema de parada de un robot.

El laboratorio donde se trabaja no tiene más de 5 puestos y las distancias de trabajo no superan en ningún momento los 15 metros. Se ha buscado que le sistema implementado se adecúe a estas especificaciones. En el caso de querer implementar un sistema comercial sería aconsejable implementar alguna de estas mejoras, ya que cubriría mayor número de necesidades de los posibles clientes.

9. Presupuesto

- Autor: Héctor Juárez Calvo.
- Departamento: Automática y electrónica.
- Descripción del proyecto: Sistema inalámbrico de parada de emergencia.
- Duración del proyecto: 5 meses.
- Desglose presupuestario:

Personal			
Apellidos y nombre	**Categoría**	**Dedicación (hombres mes)***	**Coste por mes (€)**
BARBER CASTAÑO, RAMÓN IGNACIO	Ingeniero Sénior	0,80	4.289,54
GARCÍA GODOY, DAVID	Ingeniero Sénior	0,80	4.289,54
JUÁREZ CALVO, HÉCTOR	Ingeniero	3,00	2.694,39
		Total	**14.946,434 €**

Equipos					
Descripción	**Coste (€)**	**% Uso dedicado proyecto**	**Dedicación (meses)**	**Periodo de depreciación**	**Coste imputable (€)****
PC-TOSHIBA SATELLITE	700	20	4	60	9,33
Soldador	100	10	2	60	0,33
				Total	**9,67 €**

Otras de tareas		
Descripción	**Empresa**	**Coste imputable (€)**
Impresión 3D de la carcasa de la seta	Universidad Carlos III de Madrid	0,00
·Plástico	-	6,40 €
· Utilización de la impresora	MarkeBot Industries	12,00 €
	Total	**18,40 €**

Costes de materiales				
Descripción	**Empresa**	**unidades**	**coste imputable**	**Total (€)**
Microprocesador	Arduino	7	18,5	129,50
·Entorno de desarrollo (libre)	Arduino Alpha	1	0	0,00
Módulo de radiofrecuencia XBEE	MaxStream	8	23,6	188,80
·Librería xbee.h (libre)	GNU	1	0	0,00
XbeeExplorer	sparkfun	1	20	20,00
·X-CTU (libre)	Digi	1	0	0,00
Pulsadores	Farnell	12	2,3	27,60
LEDs	Farnell	28	0,28	7,84
Interruptores	Farnell	5	3,3	16,50
Baterias	sparkfun	5	9,31	46,55
			Total	**436,79 €**

Resumen de costes	
Presupuesto Costes Totales	**Costes**
Personal	14.946,43
Amortización	9,67
Subcontratación	18,40
Costes materiales	436,79
Total	**15.410,68 €**

NOTAS:

Presupuesto estimado para un sistema con 10 setas y 2 robots.

* 1 Hombre mes = 131.25 horas. Máximo de dedicación de 12 hombres mes (1575 horas).

** Fórmula de cálculo de la amortización:

$$\frac{A}{B} x C x D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado.

B = periodo de depreciación (60 meses).

B = periodo de depreciación (60 meses).

D = % del uso que se dedica al proyecto.

Glosario

OSI	*Open System Interconnection*
ISO	*International Organization for Standardizartion*
IEEE	*Institue of Electrical and Electronics Engineers*
RF	*Radio Frecuency*
FTDI	*Future Technology Devices International*
USB	*Universal Serial Bus*
RX	*Receive*
TX	*Transmit*
WWAN	*Wireless Wide Area Network*
WMAN	*Wireless Metropolitan Area Network*
WLAN	*Wireless Local Area Network*
WPAN	*Wireless Personal Area Network*
LED	*Light Emitting Diode*
ASCII	*American Standard Code for Information*
STL	*Standard Tessellation Language*

Referencias

[1] V. Gonzalez-Pacheco, Arnaud Ramey , F. Alonso-Martin, A.Castro-Gonzalez y Miguel A. Salichs. Artículo "Maggie: A Social Robot as a Gaming Platform". De International Journal of Social Robotics Manuscript. Visualizado el 21/09/12.

[1] Descripción del modielo OSI. Wikipedia. Elace web: http://es.wikipedia.org/wiki/Modelo_OSI. Visualizado el 15/05/12.

[3] Manuel Torres Portero "*Microprocesadores y microcontroladores aplicados en la industria*". Editorial PARANINFO. ISBN: 8428316503. Fecha de publicación 1994.

[4] Página oficial de Arduino, Enlace web: http://arduino.cc/. Visualizado el 24/05/12.

[5] Byron S. Gottfried. "Programación en C", segunda edición revisada. Editorial MC GRAW HILL. ISBN:8448198468. Fecha de publicación 2005.

[6] Manual básico de programación Arduino. Enlace web: http://code.google.com/p/arduino/wiki/BuildingArduino. Visualizado el 24/05/12.

[7] Información general sistemas de radiofrecuencia de MaxStream. Página web oficial, http://www.digi.com/es/news/pressrelease?prid=500. Visualizado el 24/05/12.

[8] Kenneth Laudon y Jame Laudon. "*Sistemas de información gerencial",* décima edición. Editorial PEARSON. ISBN 9789702611912.Fecha de publicación: 2008.

[9] Manual básico de uso del software X-CTU. Página web oficial de DIGI. Enlace web: http://www.digi.com/support/productdetail?pid=3352. Visualizado el 8/07/12.

[10] Ned Moham, Tore M. Undeland y William P. Robbins "*Power Electronics"*.Editorial JOHN WILEY & SONS, INC. ISBN: 0471584088. Fecha de publicación: 2003.

[11] Andrés Barrado Bautista y Antonio Lázaro Blanco *"Problemas de electrónica de Potencia"*. Editorial PEARSON. ISBN: 9788420546520. Fecha de publicación: 2001.

[12] *".* Jorge Pleite Guerra, Ricardo Vergaz Benito y José Manuel Ruiz de Marcos "*Electrónica analógica para ingenieros"*. Editorial MC GRAW HILL. ISBN: 9788448168858. Fecha de publicación 2009.

[13] ". John Foste "Xbee CookBook Issue 1.4 for Series 1 (Freescale) with 802.15.4r. Formato Electronico. Enlace Web: http://www.jsjf.demon.co.uk/xbee/xbee.pdf. Visualizado 9/08/2012.

[14] Manual de configuración de módulos de módulos de radiofrecuencia. Pagina web oficial de arduin. Enlace web:

http://arduino.cc/en/Main/ArduinoBoardFioProgramming. Visualizado 9/08/2012.

[15] Manual de configuración de "cooking-hacks", Enlace web: http://www.cooking-hacks.com/index.php/documentation/tutorials/arduino-xbee-shield. Visualizado 9/08/2012.

[16] Página web de descarga y manual de uso de la librería *NewSoftSerial.h* de la pagina web oficial de Arduino. Enlace web: http://www.arduino.cc/en/Reference/SoftwareSerial . Visualizado 21/09 /2012.

Anexos

Índice de anexos:

A1. Hoja de características de los componentes

1.1. Arduino FIO

Arduino FIO, es un pequeño microcontrolador que está especialmente diseñado para aplicaciones inalámbricas. Funciona a 3.3 V y 8 MHz y posee 14 pines de entradas y salidas digitales y 8 entradas analógicas.

Así mismo lleva incorporado un botón de encendido y otro de reset.

También incluye un conector para incorporar una batería de litio así como un circuito de carga de la misma por medio de una conexión micro USB.

En la imagen que se muestra a continuación se muestra las medidas del Arduino Fio así como la vista superior e inferior, en la que se puede identificar todos los pines:

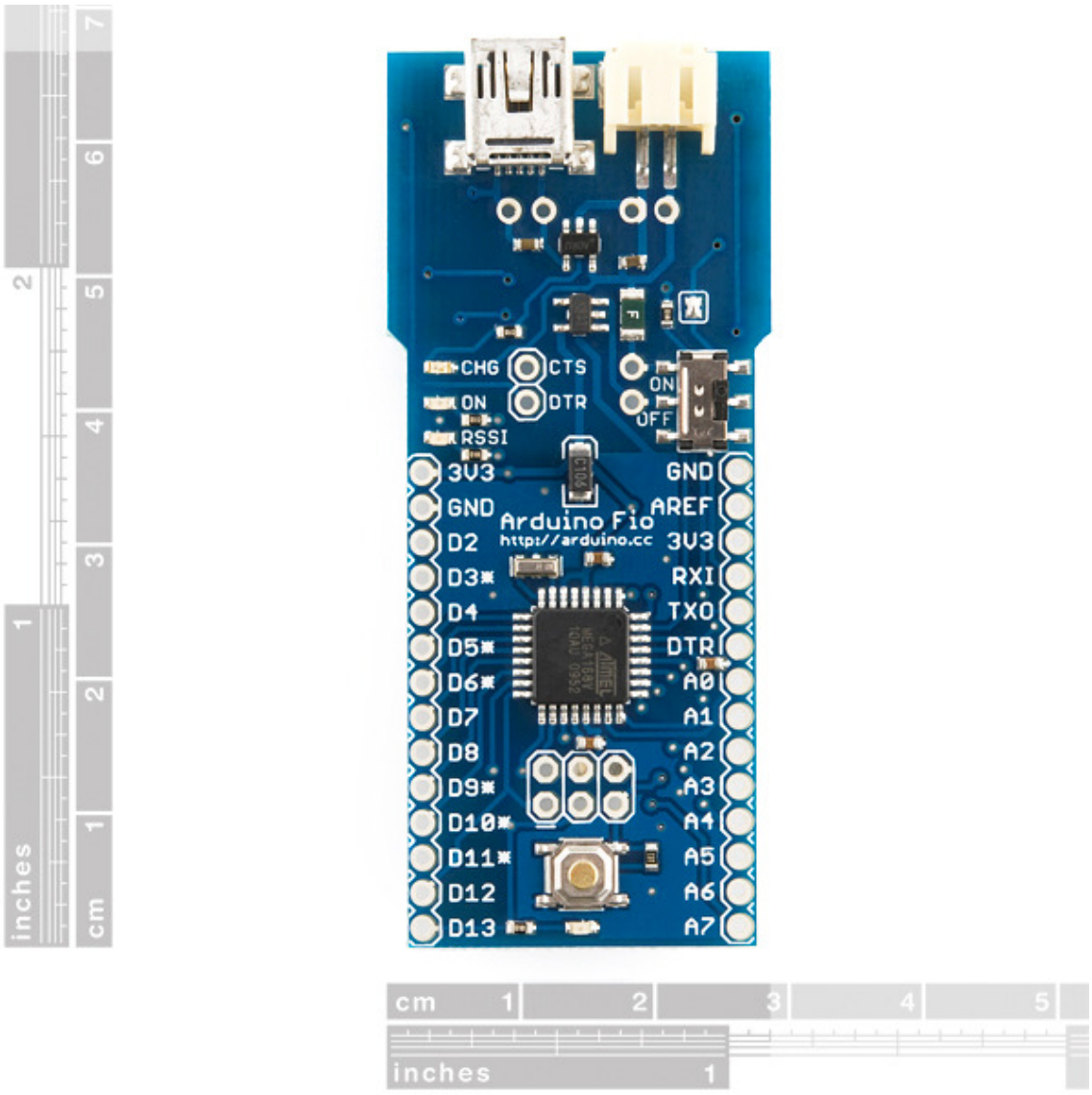


Figura 49: En esta imagen se puede observa la vista superior del Arduino fio y en ella podemos observar los pines de entrada y salida a los lados de placa, en el centro esta colocado el chip con el microcontrolador. Abajo y en la parte central nos encontramos con el botón de reset y en la parte de la derecha el botón de encendido. Finalmente en la parte superior esta situado la conexión de la batería y el circuito de carga.

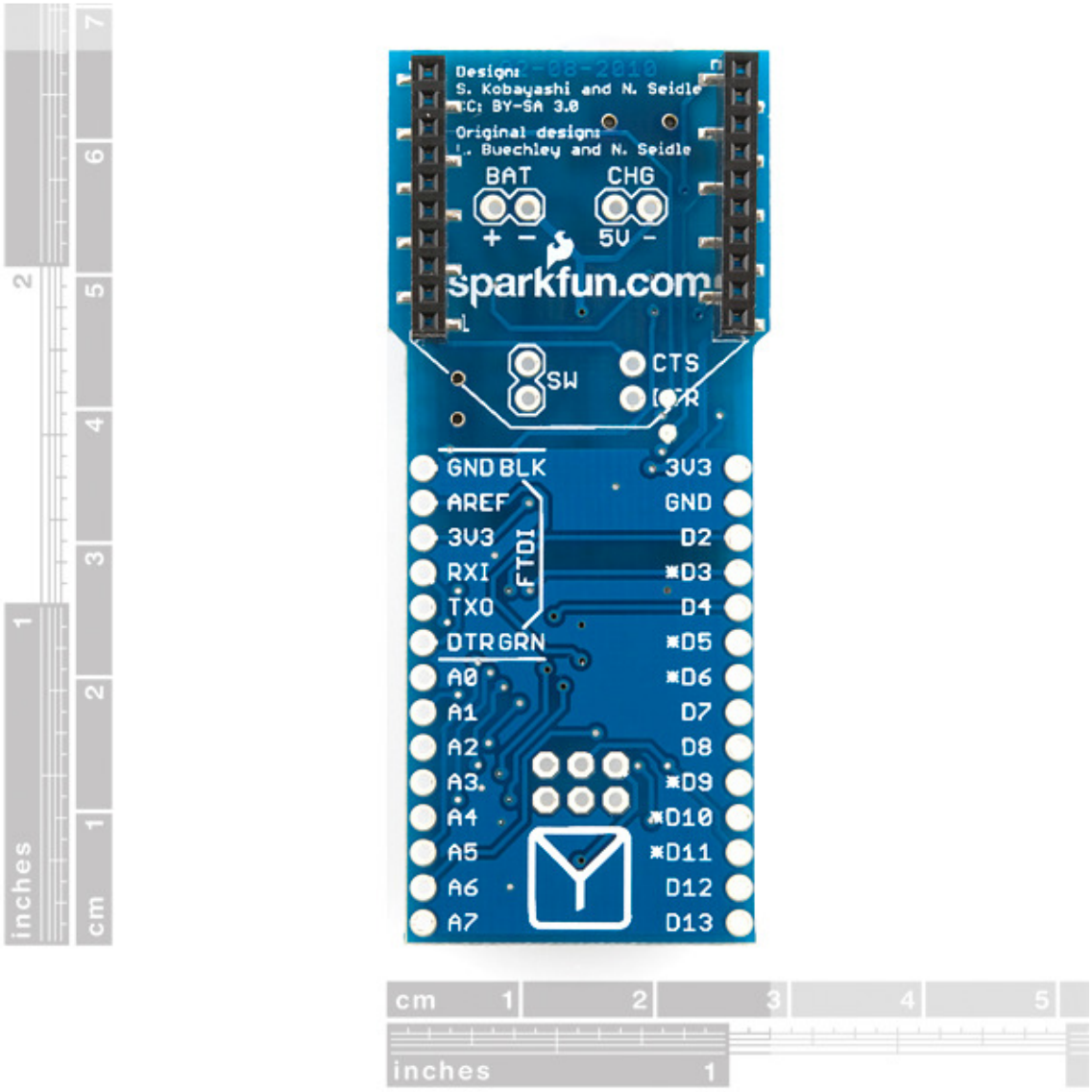


Figura 50: esta imagen representa la parte posterior del Arduino FIO así como las medidas. En ella se puede observar donde esta el zócalo para la conexión de los módulos inalámbricos. También se puede observar a los lados de la placa todos los pines.

En la imagen que se muestra a continuación se puede observar un esquema electrónico de la placa con todas las conexiones.

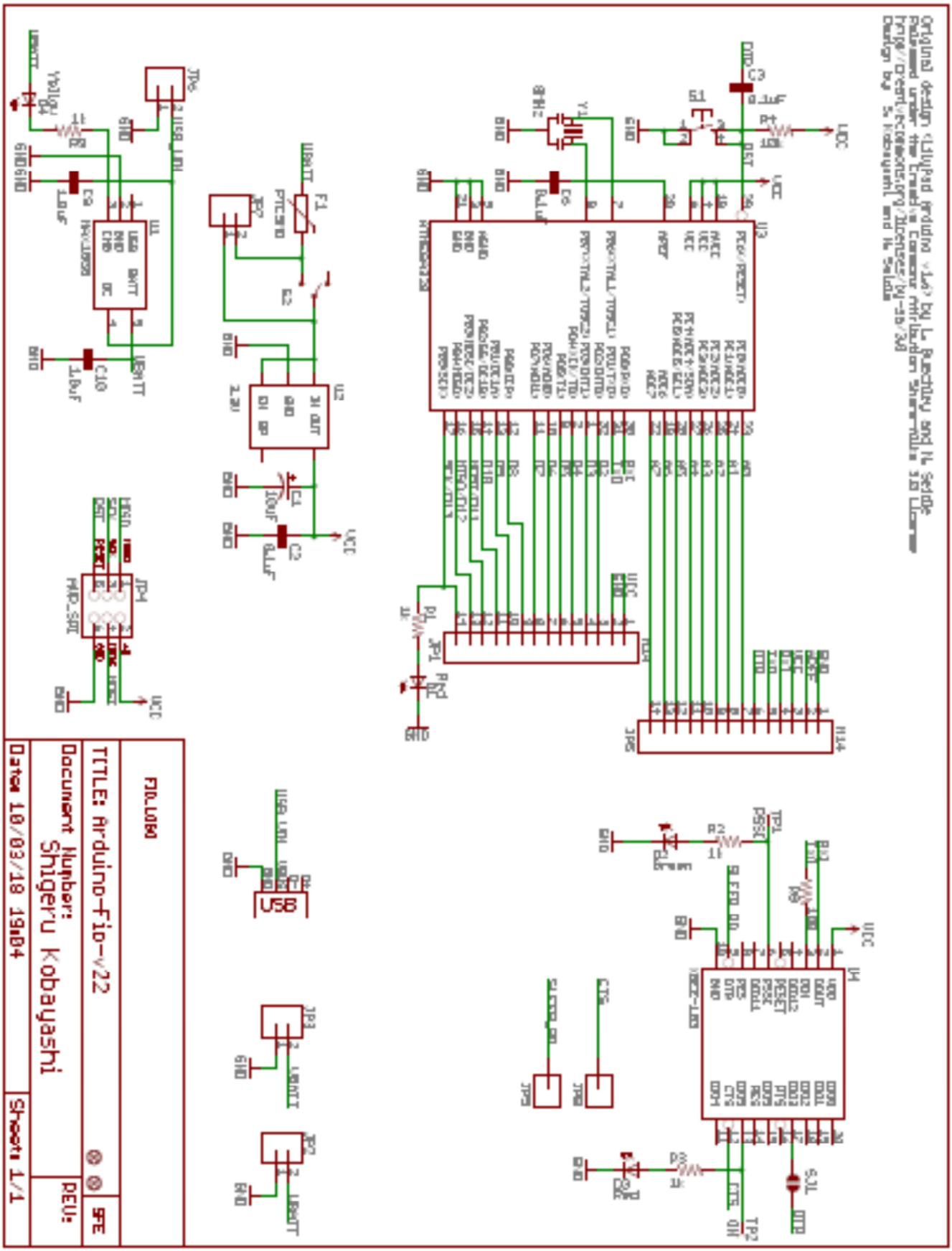


Figura 51: Esquemático Arduino FIO.

Descripción de las partes de la imagen correspondientes a los componentes hardware del Arduino FIO:

- En la vista superior izquierda se observa el esquema general de los pines de la placa así como el montaje del cristal oscilador y los botones de encendido del microprocesador.

- En la parte superior y derecha de la imagen se pueden observar todas las conexiones del zócalo para los módulos externos de radiofrecuencia con tecnología Xbee.

- Finalmente, en la parte inferior de la imagen se puede observar el circuito de alimentación de tensión para el funcionamiento de la placa y el circuito de carga de las baterías.

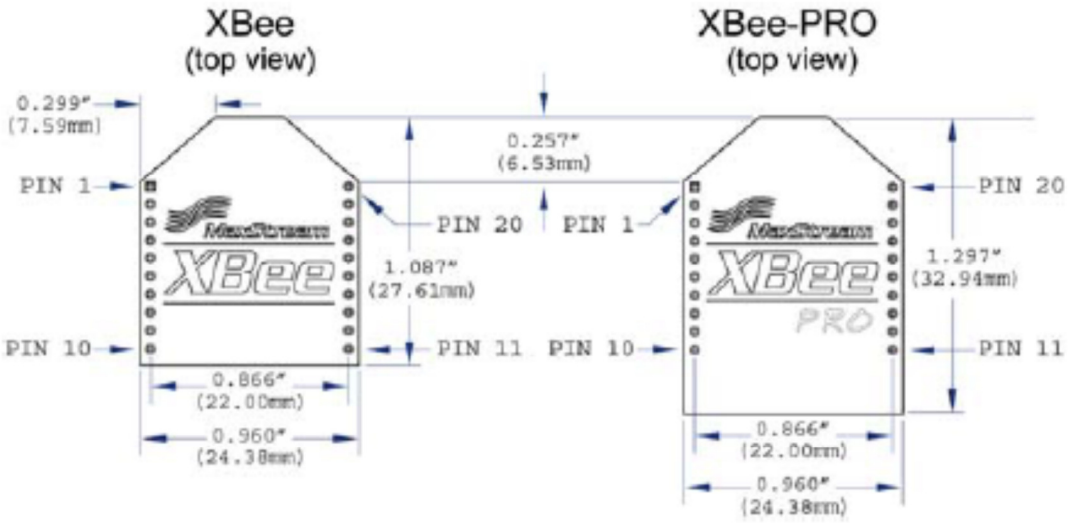
A continuación se muestra un cuadro resumen con las principales características del arduino FIO.

Microcontrolador	ATmega328P
Voltaje de trabajo	3.3 V
Voltaje de entrada	3.35 – 12 V
Voltaje de entrada en carga	3.7 – 7 V
Pines I/O Digital	14 (6 poseen salida PWM)
Pines de entrada analógica	8
Corriente Continua por los pines De entrada y salida	40mA
Memoria Flash	-32 KB (2KB dedicados al gestor de arranque) -2 KB de SRAM -1 KB de EEPROM
Pines de alimentación	-**BAT**: alimentar la placa con una batería de litio -**CHG**: terminales de carga -**3V3**: pines de alimentación regulada a 3.3 V -**GND**: pines de masa
Entradas y salidas	-Serie: **RXI (D1) y TXO (D1)** -Interrupciones externas: **D2** y **D3** -PWM: **D3, D5, D6, D9, D10** Y **D11** -D13 dispone de LED incorporado -Comunicación SPI: **D10** (SS), **D11** (MOSI), **D12** (MISO) Y **D13** (SCK) -Comunicación I_2C: **D4** (SDA) y **D5** (SCL) -Voltaje de referencia para entradas analógicas: **AREF** -Reinicio del microcontrolador: **DTR**

Tabla 2: Principales características Arduino FIO

1.2. Modulo de radio frecuencia XBee

A continuación se muestra el esquemático (Figura 52) junto con las medidas y una pequeña tabla con las principales características del XBEE SERIE 1 PRO que ira alojado en la placa Arduino FIO:



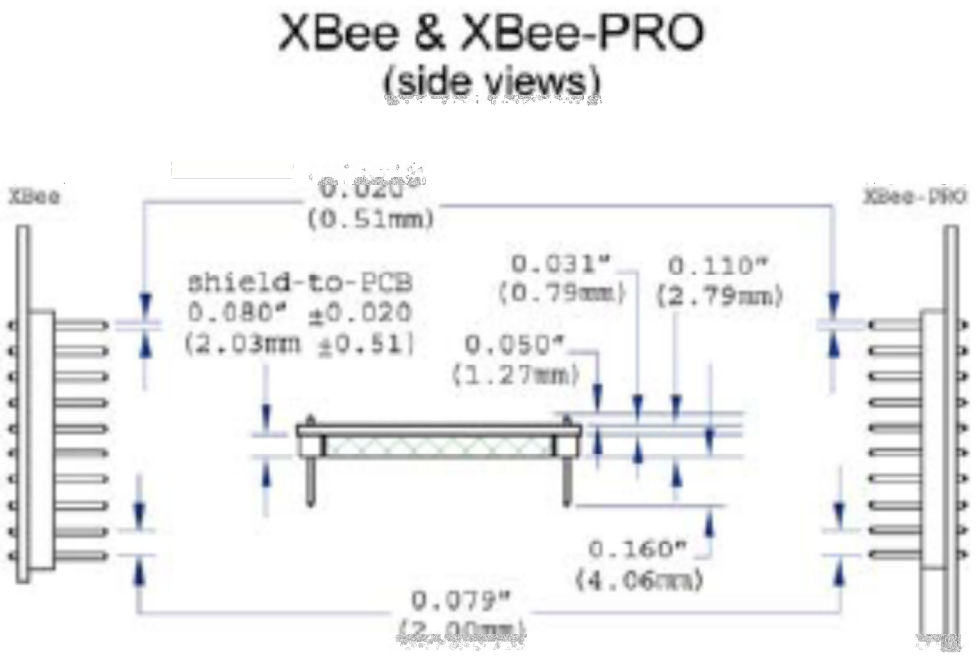


Figura 52: Esquemático de los módulos de radiofrecuencia XBEE.

Fabricante	MaxStream
Tensión de alimentación	3.3 V
Corriente máxima	215 mA
Frecuencia	2.4 GHz
Pila	802.15.4 con protocolo serie (hace uso de TX y RX del microprocesador
Velocidad de transferencia máxima	250 kbps
Potencia de salida	60 mW (+18 dBm)
Alcance	1500 m
Otras características	-Antena intgrada -Certificado FCC.6 pines ADC de 10-bit -8 pines digitales IO -Encriptación 128-bit -Configuración local o de forma inalámbrica

Tabla 3: principales caraterísticas XBEE PRO

1.3. Xbee Explorer

El XBee Explorer USB permite conectar y utilizar cualquier módulo XBee directamente mediante un puerto USB.

También puede utilizarse para configurar los módulos, para un uso posterior en cualquier comunicación inalámbrica entre Xbee.

A continuación se presenta un esquema eléctrico del Xbee Explorer.

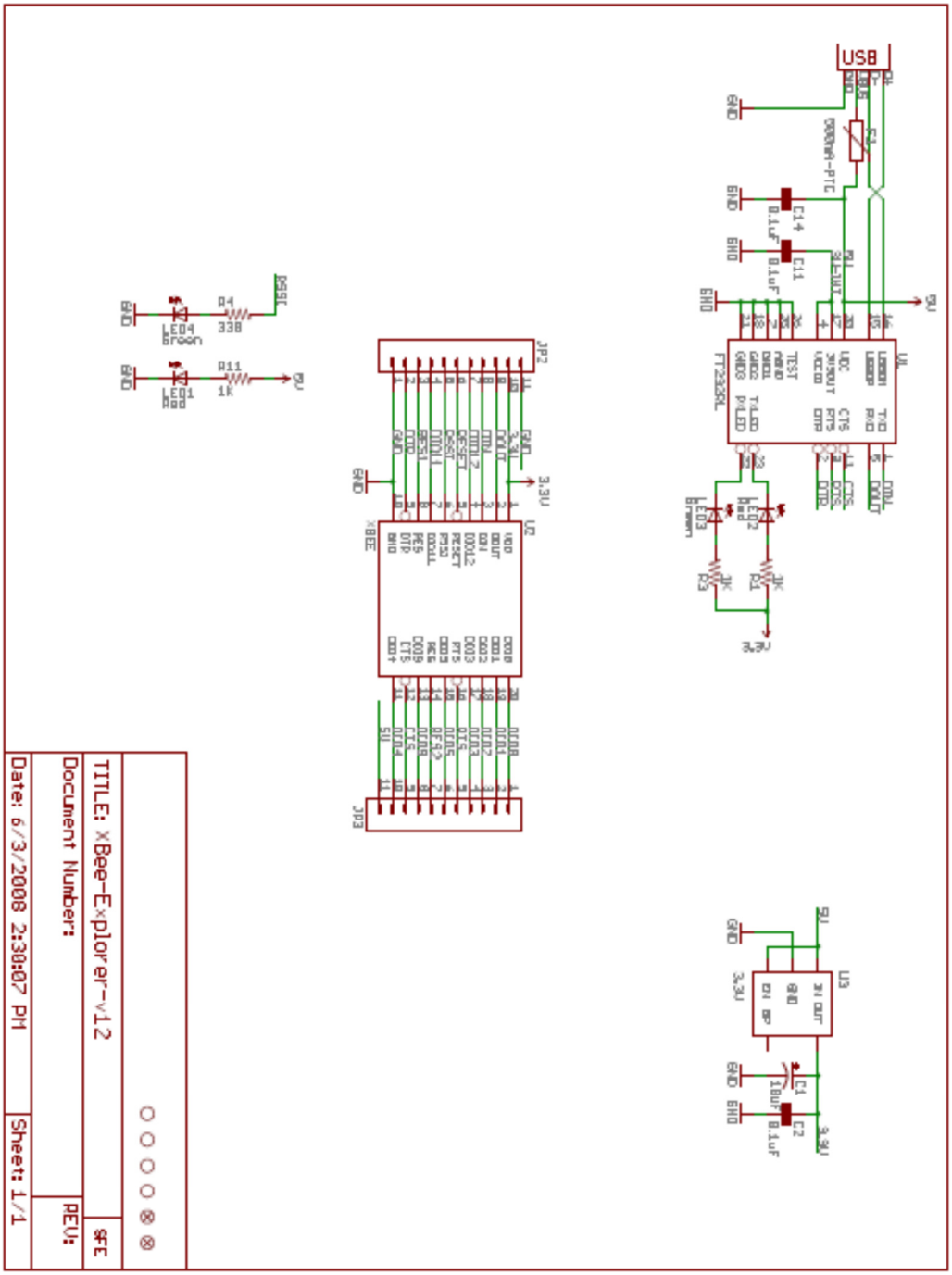


Figura 53: Esquema eléctrico Xbee Explorer.

1.4. Bateria

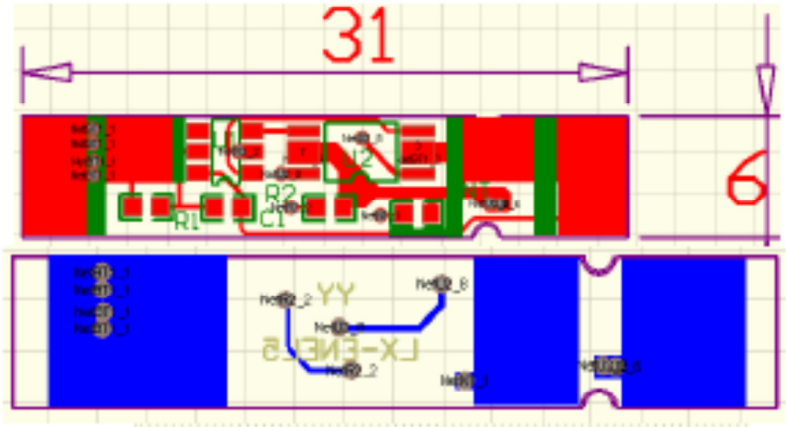
Batería Lipo (Polymer Lithium) de alta calidad y capacidad, proporciona 900mAh a una tensión de 3.7V. Este tipo de baterías proporciona una fuente

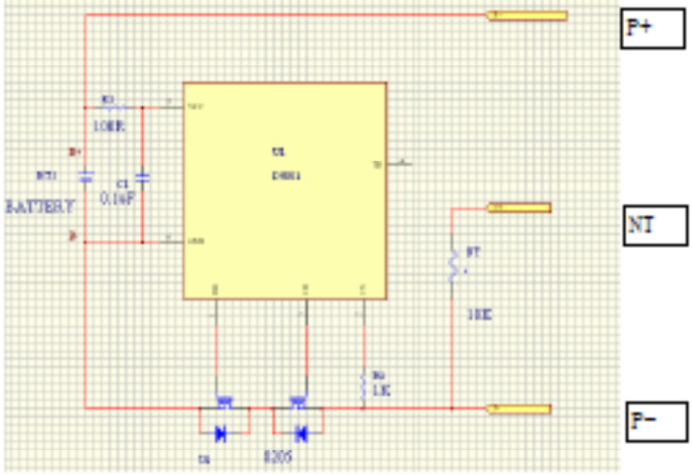
de alimentación muy eficiente, a la vez que son muy ligeras y soportan altos picos de corriente. En la Figura 54 puede observarse sus medidas.

Además incorpora un conector tipo Molex para conectarlos directamente a un circuito al cargado USB. Se puede observar todo el esquema eléctrico en la Figura 55

Figura 54: Dimensiones Bateria LIPO

Title	PCM specification	Page	1 of 1





Pin Layout Diagram	
Pin	Description
B+	Connect to positive terminal of cell
B-	Connect to negative terminal of cell
P+	Connect to Camera +/ charger +
NT	10K Ω resistance
P-	Connect to Camera -/ charger -

Key Components:

Item	P/N	Maker
Protection IC	DW01+	FORTUNE or equiv
MOSFET	SS8205	AOS or equiv

Specification of PCB

Material	FR-4
Dimension	31*6*0.5mm
UL	94-V0

Figura 55: esquema de conexión de la batería LIPO

1.5. LEDs

Los LEDs elegidos son pequeños LEDs (Figura 56) para circuitos de corriente continua de baja potencia y con bajo consumo. A continuación se muestra una tabla con las principales características.

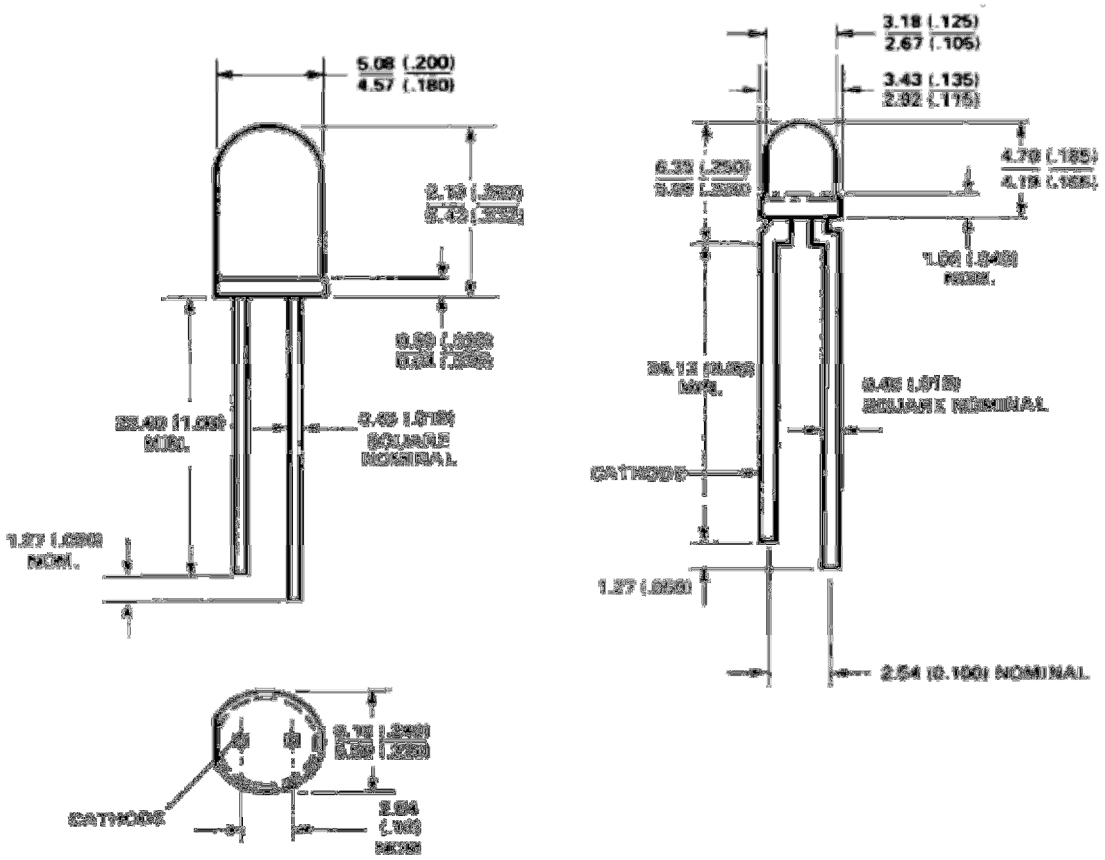


Figura 56: Tamaño de los LEDs

Tamaño de la capsula	T – 1 (3mm)
Intensidad luminosa	2.1 mcd
Ángulo de visión	50º
Corriente de funcionamiento	3 mA
Caida de tensión	1.8 V
Rango de temperatura	-55 a 100 ºC
Pico de corriente máxima	7 mA
Pico de tensión máxima	5 V

Tabla 4: Características de los LEDs

1.6. Pulsadores

A continuación se muestra una imagen en la que se puede observar el tamaño y una tabla con las principales características de los pulsadores para el arranque y parada del robot.

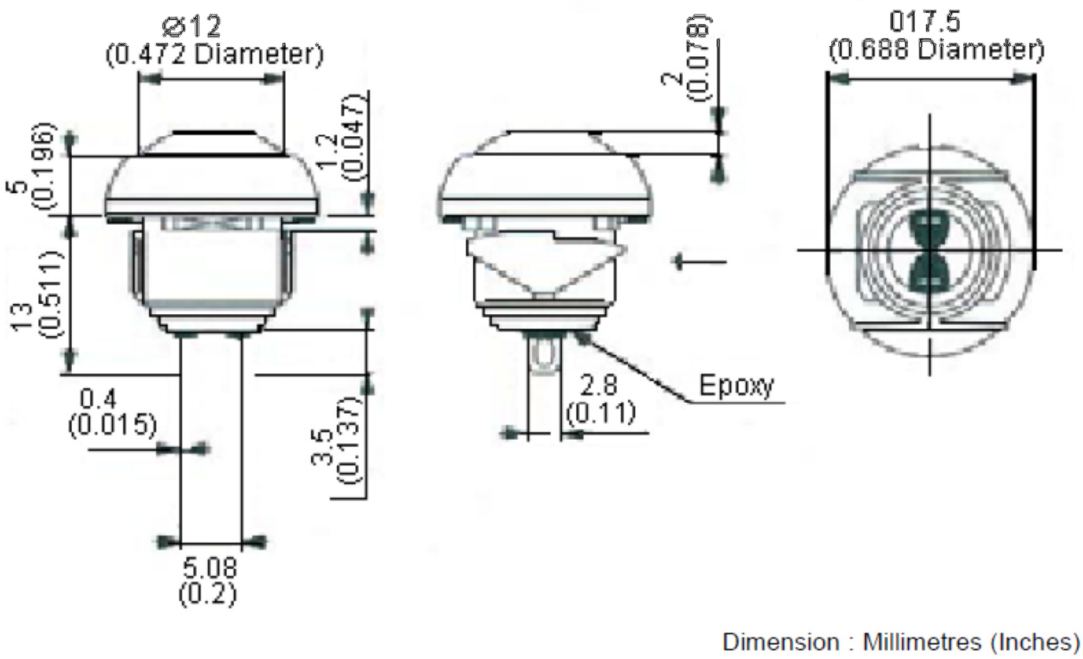


Figura 57: Tamaño de los pulsadores de encendido y parada del robot

Configuración de los contactos	SPST
Tensión máxima admisible en corriente alterna	125 V
Tensión máxima admisible en corriente continua	50 V
Corriente máxima	400 mA
Modo de enlace	Soldadura

Tabla 5: Características de los pulsadores

1.7. Transistor

Los transistores se usan para poder controlar los relés que activan los motores de los robots, ya que la corriente proporcionada por el micro no es suficiente para activarlos. El chip elegido es el ULN2065B (ver Tabla 6) el cual incorpora 4 transistores con los diodos de libre circulación, tal como se muestra en la siguiente figura.

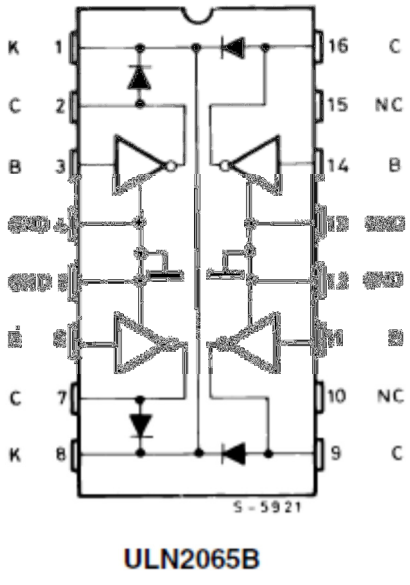


Figura 58: Esquema del ULN2065B

Polaridad	NPN
Tensión colector - emisor	80 V
Potencia disipada	4.3 W
Corriente de colector	1.5 A
Número de pins	16
Tensión de entrada mínima para funcionamiento	2 V
Temperatura de funcionamiento	-20 a 85 ºC

Tabla 6: Principales características del transistor ULN2065B.

A2. Configuración de los módulos XBEE para la transmisión del código del programa al microprocesador

-Paso 1: Configuración del módulo que transmite la información:

Este módulo estará acoplado al Xbee Explorer (Figura 59). Para ello se conectara el modulo Xbee con el Xbee Explorer y este con el ordenador mediante un cable USB. El ordenador detectará el Xbee Explorer COM puerto serie y se le adjudicará un puerto COM, por ejemplo el COM4.

Figura 59 Xbee explrer.

A continuación encendemos el programa X-CTU y señalamos el puerto COM con el que (Figura 60) .

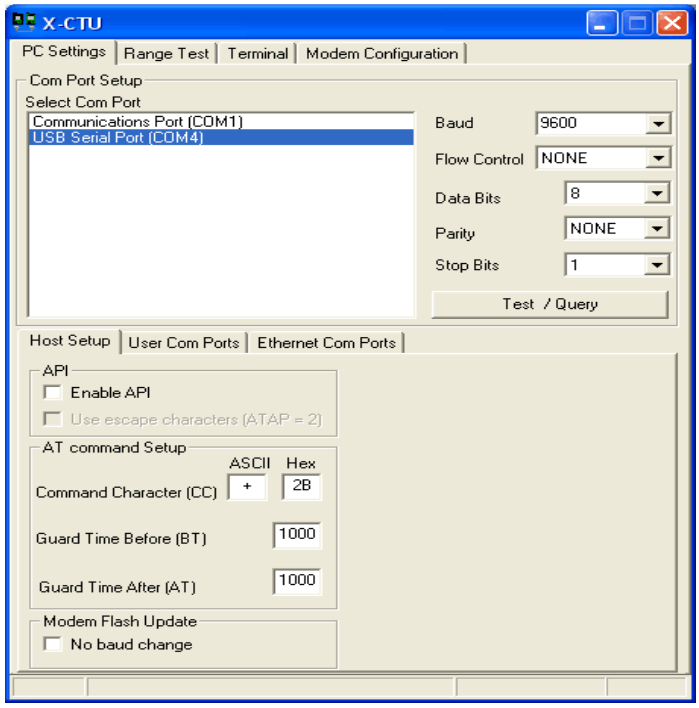


Figura 60.

Una vez conectado y verificada la conexión y comunicación entre ordenador y Xbee Explorer, pasamos a configurar el módulo Xbee, para ello pulsaremos "Read" en la pestaña de "Modem Configuration".

Primeramente seleccionaremos la PAN ID (Figura 61), es un digito de 4 números que identifica a l red que conectará a los dos módulos, por ejemplo 3137.

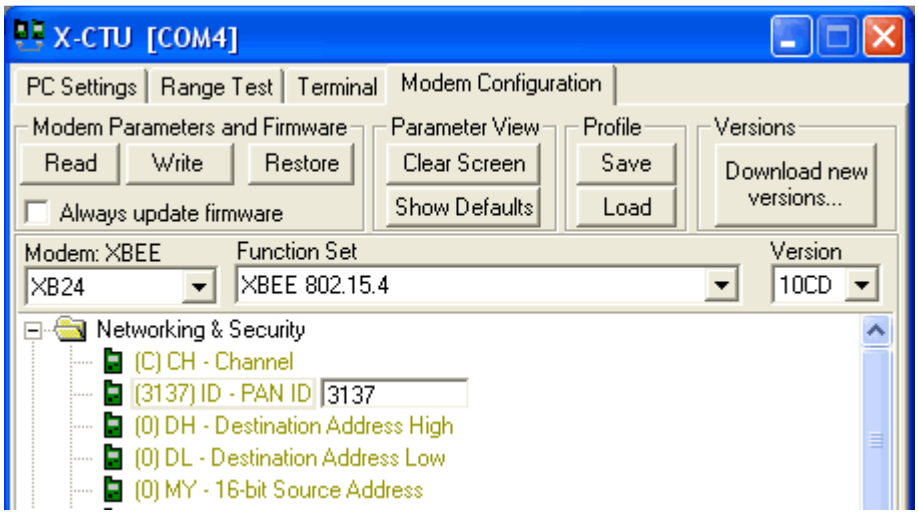


Figura 61.

Seguidamente seleccionamos la velocidad de transmisión inalámbrica (Figura 62), por el tipo de chip ('328p nuevo) usado será necesario que la velocidad de transmisión se de 57600 baudios (sería 19200 si utilizaramos el chip 168 ó '328p antiguo)

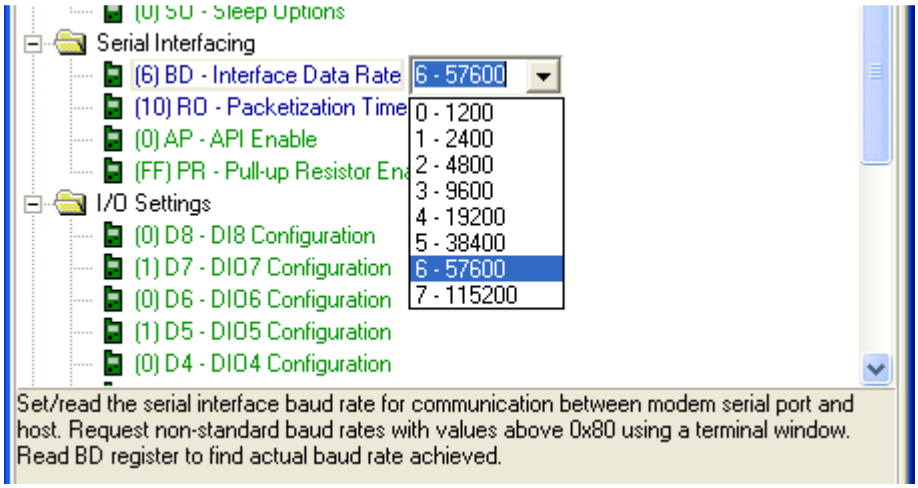


Figura 62.

A continuación determinaremos el tiempo de espera antes de la transmisión del código para que la recepción de los paquetes sea correcta. Habra que modificar "Packetization Timeout" (Figura 63) y ponerlo a 10, ya que se transmitirán códigos con un peso inferior a 10K.

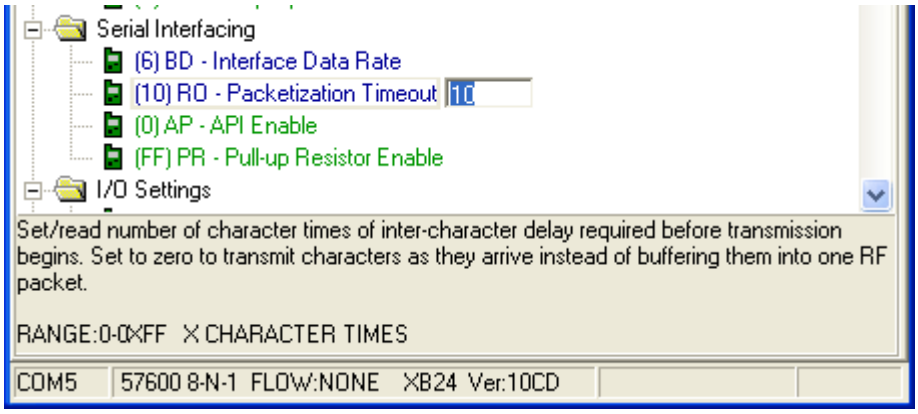


Figura 63.

Configuramos el pin 3 como entrada digital (Figura 64).

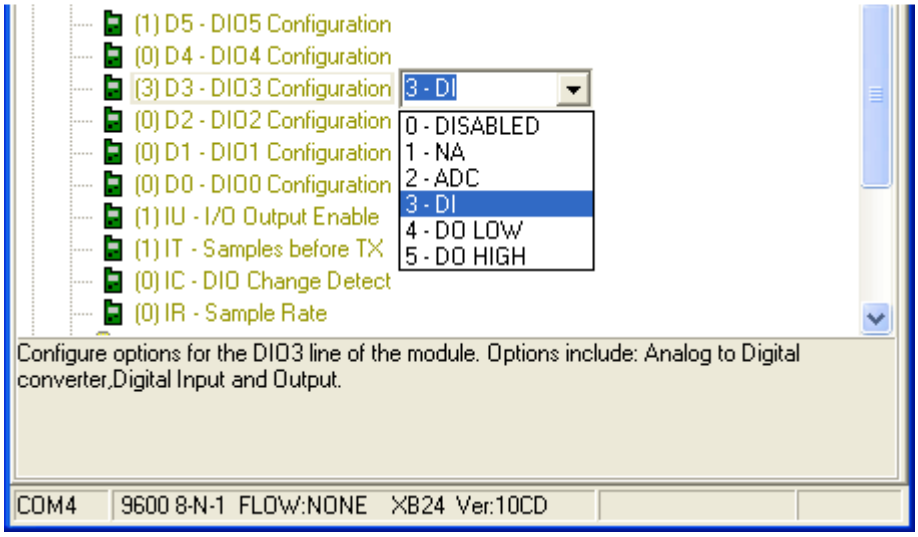


Figura 64.

Ajustamos "Digital IO Change Detect" a FF (Figura 65), para que la lectura del puerto D3 sea la correcta y pulsamos "Write" para que toda la información quede modificada.

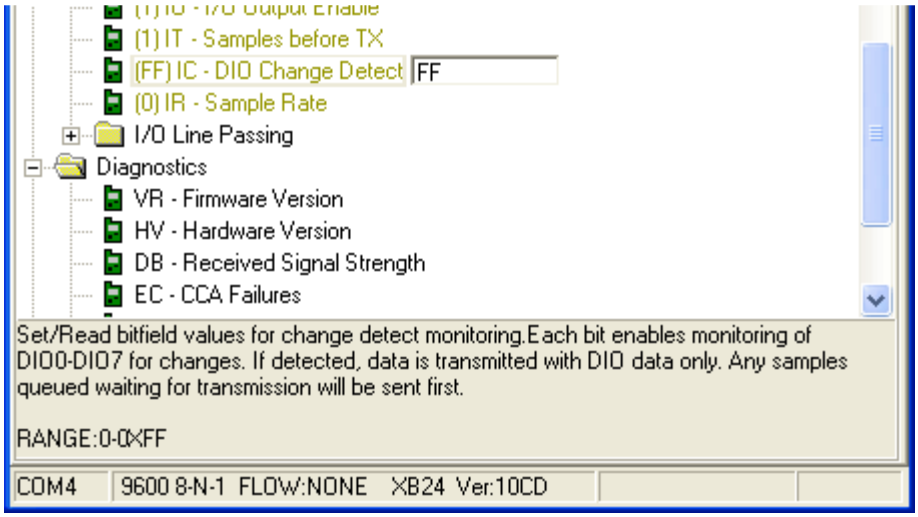


Figura 65.

Finalmente se conmutará la patilla pin D3 y el RTS de la placa Xbee Explorer y se marcara la opción Set RTS On Close (Figura 66). Este menú lo encontraremos *en panel de control → Hardware → Administrador de dispositivos → Puertos (COM y LPT).*

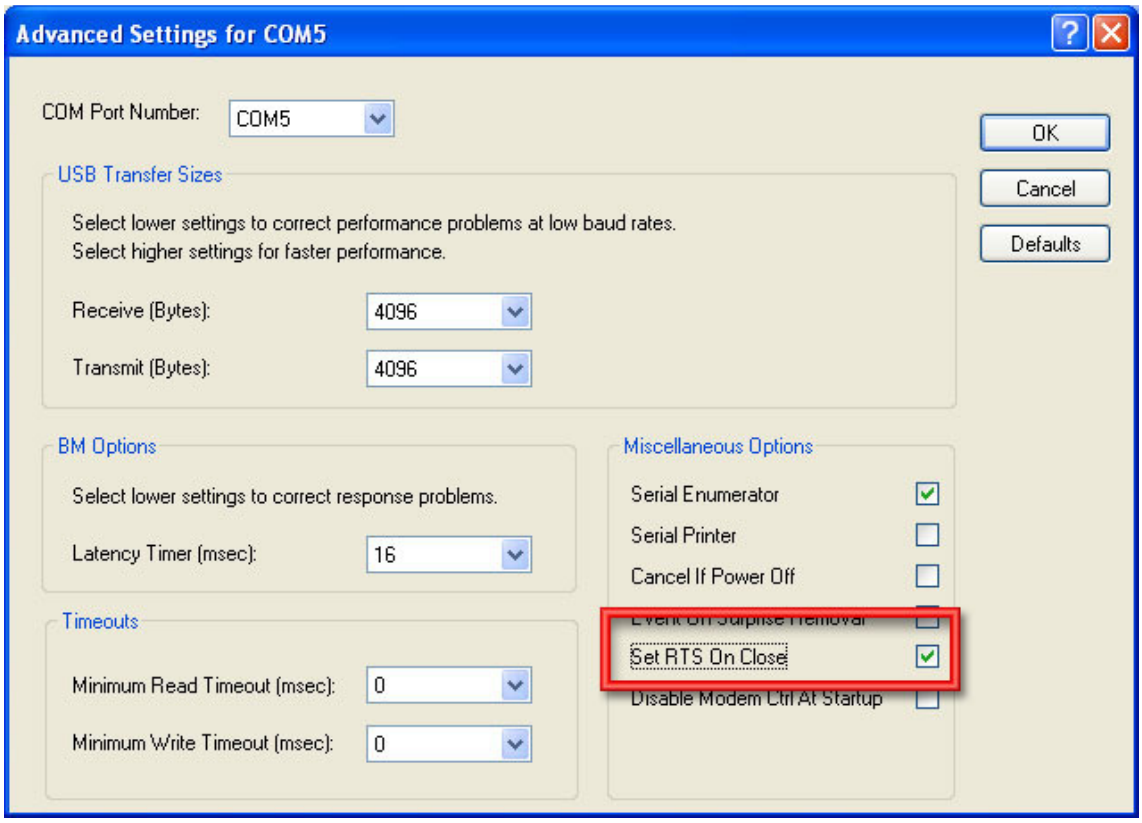


Figura 66.

-Paso 2: Configuración del módulo que recibe la información:

Conectamos el otro módulo Xbee al Xbee Explorer y este al ordenador mediante el cable USB.

Como en el paso anterior abrimos el X-CTU y leemos la configuración del Xbee en la pestaña “Modem Configuration”.

A continuación modificamos “PAN ID” (Figura 67) con la misma que tenia el modulo emisor.

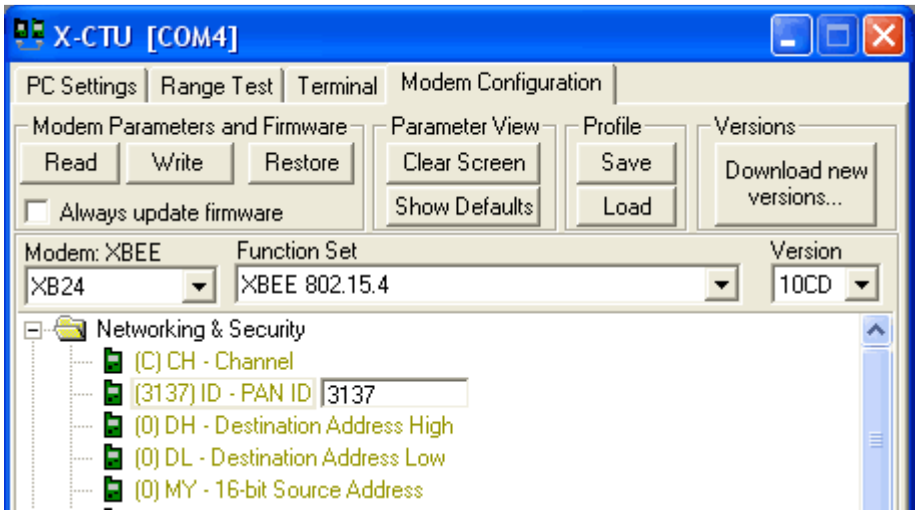


Figura 67.

Ajustamos la velocidad de transmisión (Figura 68) a la misma que el otro Xbee, es decir en nuestro caso a 57600 baudios.

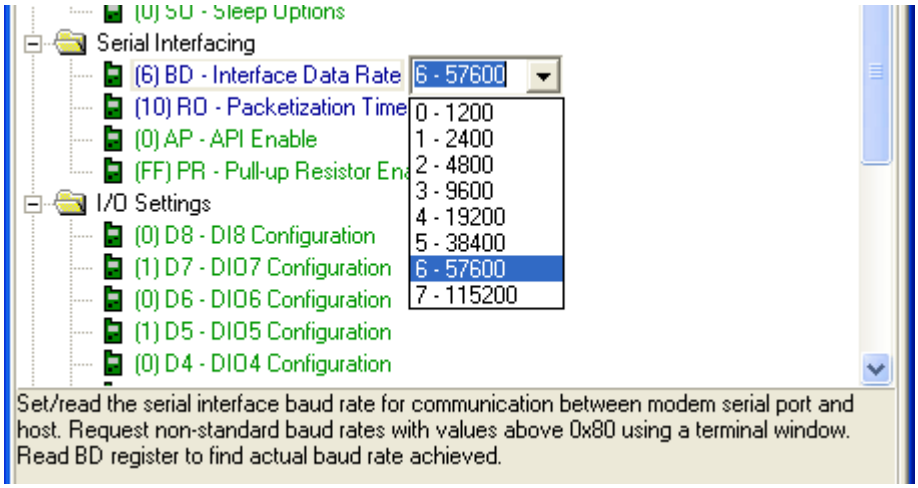


Figura 68.

Como en el caso anterior ajustamos la opción de Packetization Timeout a 10 (Figura 69).

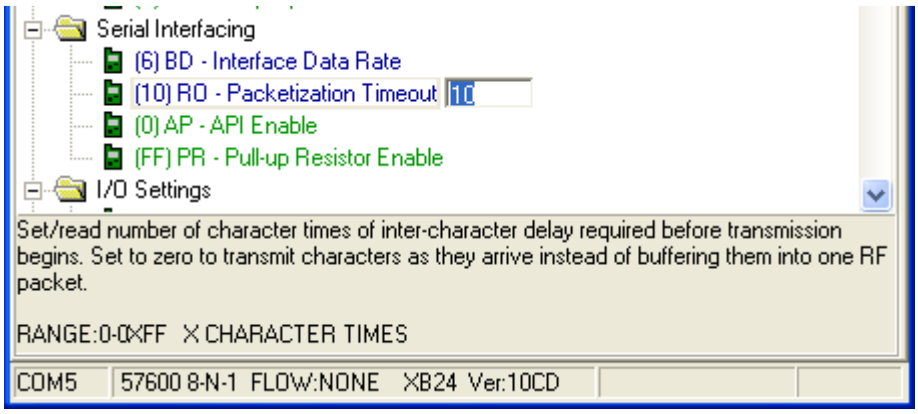


Figura 69.

A continuación configuramos el pin D3 como salida digital (Figura 70), y que esté por defecto a nivel alto.

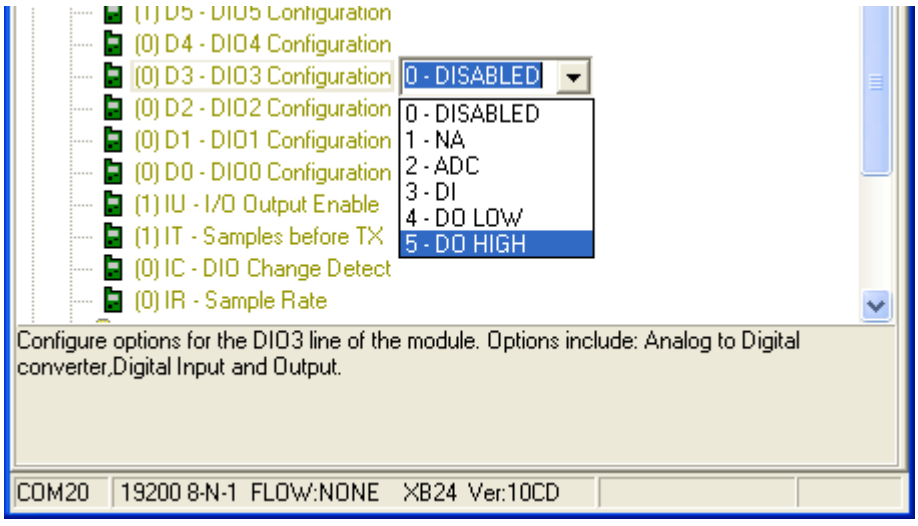


Figura 70.

Ajustamos la opción I/O Output Enable a 0 (desabilitada) (Figura 71). Evita que el receptor actualice su estado, sin tener en cuenta la configuración que se le dio al pin D3 para que la comunicación sea la correcta.

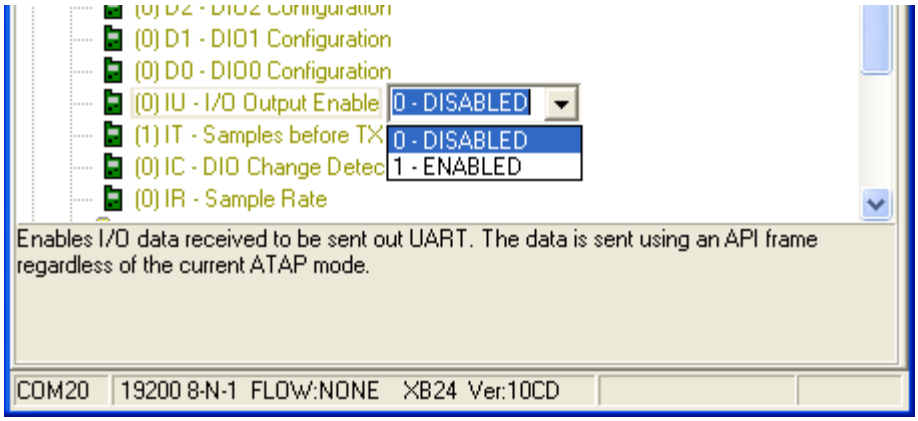


Figura 71.

Finalmente hay que poner I/O Input Adress a FFFF (Figura 72). Se pone FFFF por simplicidad y para que pueda leer de cualquier dirección, sino habría que haber grabado en el emisor y en el receptor la misma dirección.

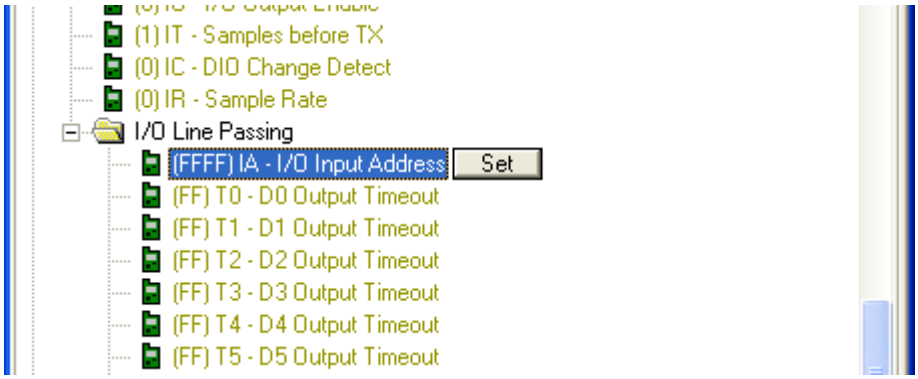


Figura 72.

Nota: a la hora de cargar el programa se realizará el procedimiento estándar, como si la transmisión fura por un puerto serie normal.

A3. Configuración Xbee para la comunicación inalámbrica del sistema

3.1. Configuración Xbee alojado en el robot

Cada robot generará una red diferente, por lo que el principal aspecto a tener en cuenta es saber las PAN ID que se le dará a cada robot y que a cada robot se le grabe una PAN ID diferente.

El método seguido es el siguiente, como tenemos la posibilidad de controlar 8 robots diferentes las pan ID irán desde 3330 hasta las 3337. Para ello haremos uso del programa X-CTU (Figura 73)

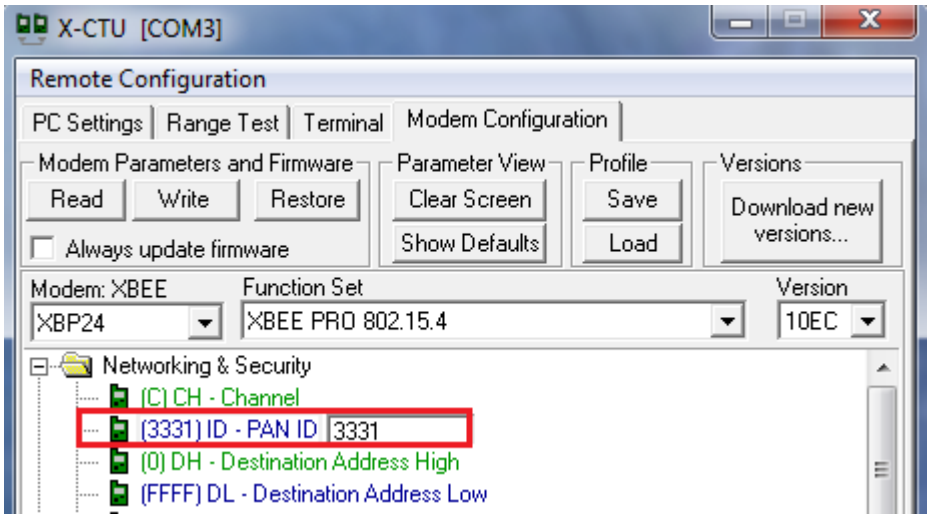


Figura 73.

A continuación le daremos la dirección a la que se enviarán todos los paquetes. Este parámetro será el mismo para todos los robots, y la dirección será la 0x1. Para modificar esta dirección se cambiará el parámetro MY (Figura 74).

La dirección a la que queremos enviar no hace falta configurarla con xbee, ya que desde la librería xbee.h podemos elegir a que dirección queremos enviar el paquete. Para el caso del robot, lo que hará es un broadcast, es decir la dirección será la 0xFFFF.

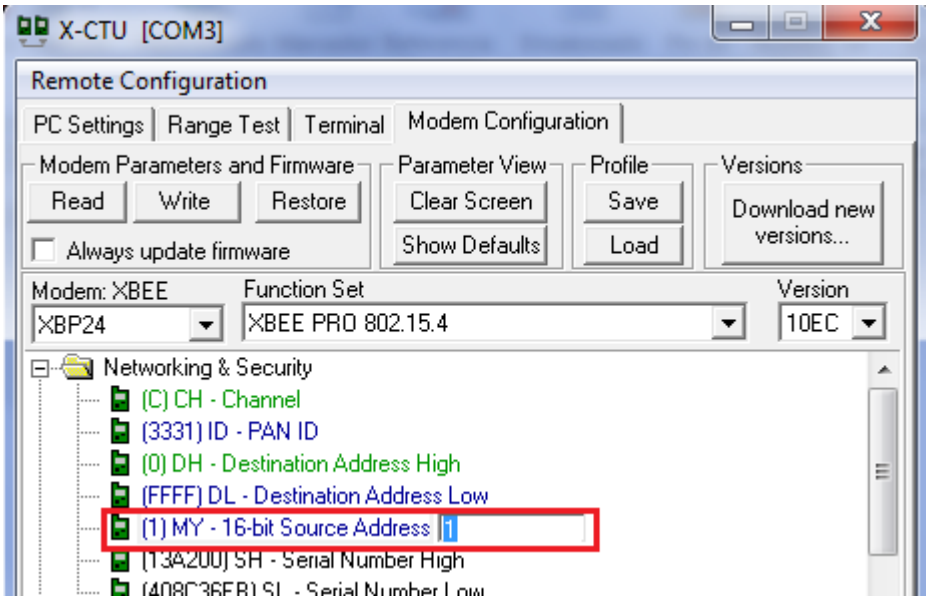


Figura 74.

El siguiente paso es elegir la velocidad de transmisión, esta será de 57600 baudios. Para ello cambiaremos el parámetro BD a 57600 (Figura 75).

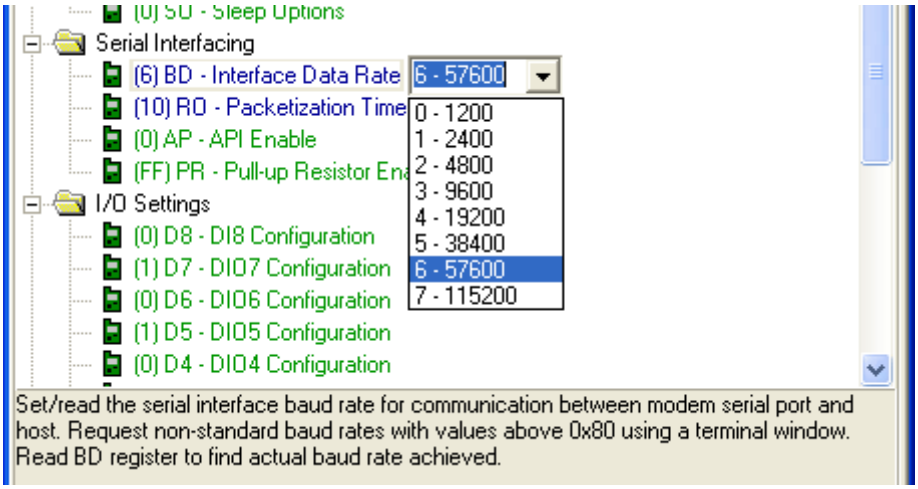


Figura 75.

Para poder usar la librería xbee.h será necesario estar en modo de funcionamiento API. Para ello ajustaremos el parámetro AP a 2 (Figura 76).

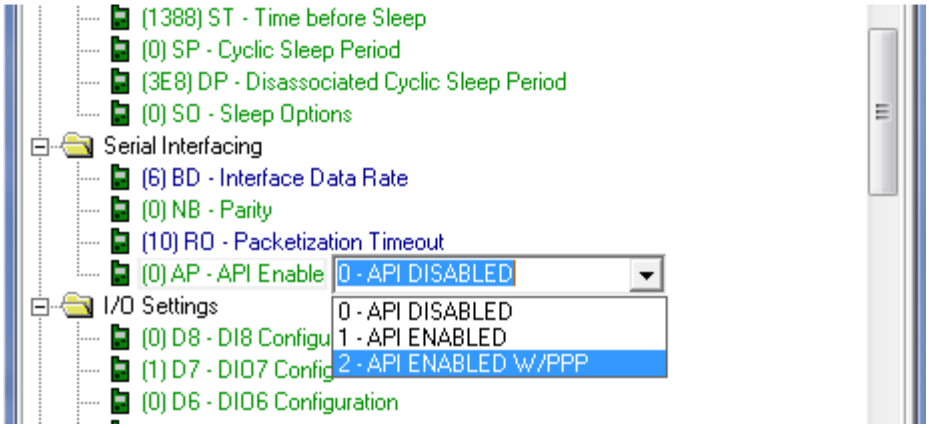


Figura 76

Una vez realizamos todos los cambios pulsamos el icono “write” y ya tenemos configurada la parte del robot.

3.2. Configuración Xbee alojado en la seta

A diferencia que el robot, la seta será la que elegirá con que robot se quiere conectar, por lo que no será necesaria configurar aquí la PAN ID del robot, se configurara por medio de comando en el setup() del programa (ver anexo del código de la seta).

Para las setas tenemos que tener la consideración de darle una dirección diferente, esta dirección empezará en la 0x2 hasta 0xC ya que el sistema esta diseñado para soportar a 10 setas como máximo, y la dirección 0x1 está reservada para los robots. Para ello ajustamos el parámetro MY con el programa X-CTU. (Figura 77).

La dirección de destino será la 0x1 y será para todos los robots, se puede establecer con X-CTU o mediante el uso de la librería xbee.h.

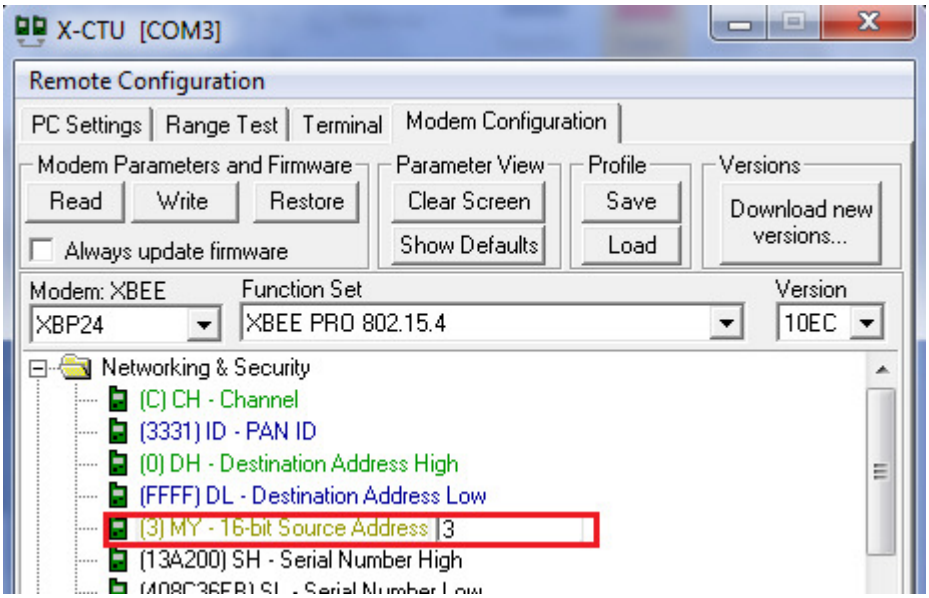


Figura 77.

A continuación ajustamos la frecuencia de transmisión a 57600 baudios modificando el parámetro BD (Figura 78).

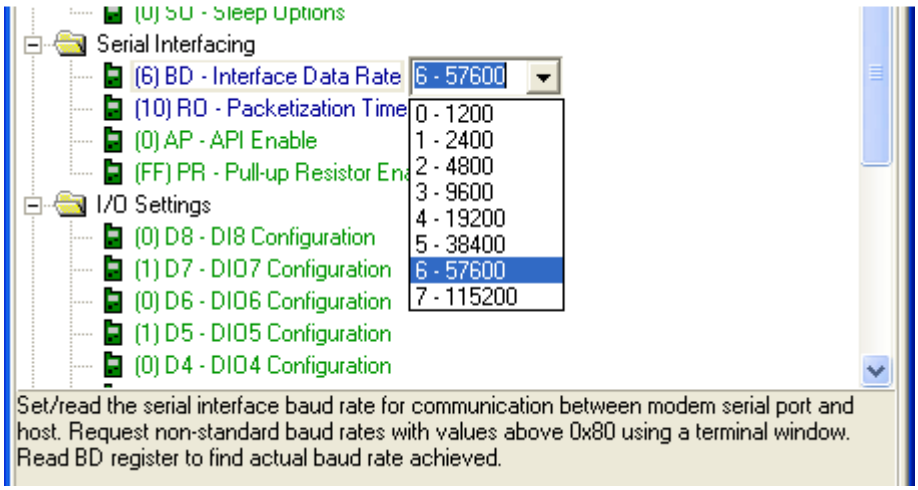


Figura 78.

Finalmente como en el caso de configuración del Xbee alojado en la seta será necesario ajustar el modo de funcionamiento API para poder hacer uso de las librerías xbee.h. Para ello ajustamos el parámetro AP a 2 (Figura 79).

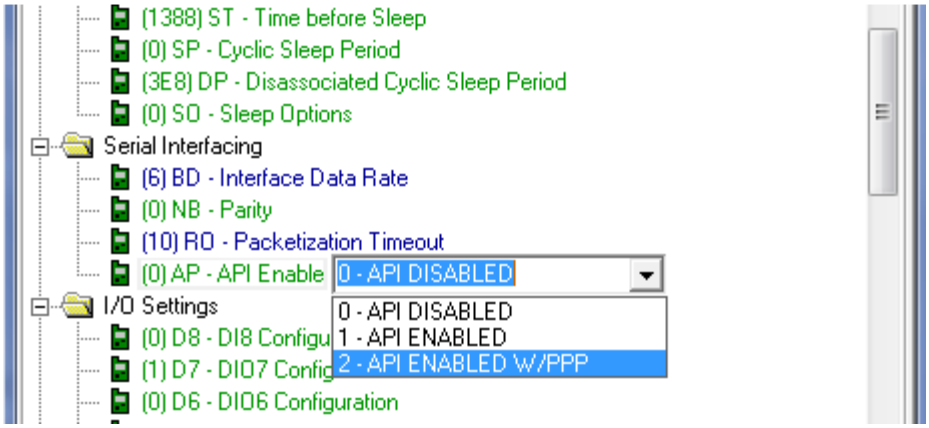


Figura 79.

El último paso es pulsar el icono “write” para guarda la configuración.

A4. Datos obtenidos de la prueba de medición de la curva de descarga de la batería

4.1. Mínima duración de la batería

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
643	0	641	785104	618	1565888	614	2346700
643	34404	622	815134	632	1595920	630	2376732
642	64428	640	845164	636	1625952	633	2406764
627	94455	636	875194	617	1655980	632	2436796
645	124483	636	905220	636	1686011	631	2466828
626	154513	636	935249	633	1716043	628	2496856
641	184539	636	965279	634	1746075	628	2526885
642	214568	636	995309	636	1776107	632	2556917
625	244598	636	1025335	617	1806139	630	2586945
625	274624	638	1055363	632	1836171	630	2616977
625	304654	635	1085392	632	1866199	628	2647009
640	334680	639	1115424	631	1896230	628	2677037
639	364707	635	1145456	616	1926262	613	2707066
639	394733	635	1175488	616	1956294	628	2737098
639	424763	638	1205516	616	1986326	631	2767130
639	454793	635	1235544	631	2016358	627	2797162
639	484823	636	1265573	634	2046390	630	2827194
638	514848	634	1295605	630	2076418	630	2857222
638	544878	634	1325637	615	2106449	612	2887254
638	574908	635	1355669	634	2136481	629	2917285
623	604934	619	1385701	630	2166513	629	2947317
638	634963	619	1415729	634	2196545	627	2977349
638	664993	634	1445761	631	2226577	626	3007381
622	695019	618	1475792	614	2256609	630	3037413
640	725049	634	1505824	633	2286637	611	3067445
623	755075	637	1535856	633	2316668	630	3097473

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
614	2346700	609	3698102	619	5049499	597	6400897
630	2376732	624	3728130	618	5079531	612	6430928
633	2406764	624	3758159	618	5109563	598	6460960
632	2436796	609	3788191	618	5139595	612	6490992
631	2466828	628	3818219	603	5169623	612	6521024
628	2496856	624	3848251	619	5199654	611	6551056
628	2526885	627	3878283	621	5229686	597	6581088
632	2556917	609	3908311	603	5259718	596	6611116
630	2586945	628	3938340	617	5289750	596	6641147
630	2616977	609	3968372	602	5319782	615	6671179
628	2647009	624	3998404	617	5349814	596	6701211
628	2677037	627	4028436	602	5379842	596	6731243
613	2707066	624	4058468	621	5409871	597	6761275
628	2737098	623	4088496	617	5439903	595	6791307
631	2767130	623	4118528	616	5469935	612	6821335
627	2797162	623	4148559	616	5499963	610	6851364
630	2827194	626	4178591	601	5529995	610	6881396
630	2857222	623	4208623	620	5560023	613	6911424
612	2887254	608	4238655	601	5590052	611	6941456
629	2917285	608	4268687	620	5620084	610	6971488
629	2947317	622	4298715	619	5650116	610	7001516
627	2977349	626	4328747	600	5680148	610	7031545
626	3007381	624	4358778	615	5710180	610	7061577
630	3037413	626	4388810	619	5740208	610	7091609
611	3067445	622	4418842	615	5770240	613	7121641
630	3097473	622	4448874	615	5800271	613	7151673
627	3127504	622	4478906	616	5830303	609	7181701
626	3157536	625	4508938	615	5860335	610	7211732
628	3187568	607	4538966	599	5890367	612	7241764
626	3217600	625	4568997	617	5920399	609	7271796
625	3247632	625	4599029	614	5950427	609	7301828
611	3277664	607	4629061	617	5980459	609	7331860
627	3307692	621	4659093	614	6010490	594	7361892
625	3337723	620	4689125	614	6040522	593	7391924
610	3367755	620	4719157	598	6070554	612	7421952
625	3397787	620	4749185	598	6100586	608	7451983
626	3427819	624	4779216	617	6130618	608	7482015
629	3457851	624	4809248	613	6160650	609	7512047
626	3487883	620	4839280	616	6190678	594	7542079
625	3517911	605	4869312	617	6220709	608	7572111
610	3547942	619	4899344	614	6250741	593	7602143
610	3577974	620	4929376	616	6280773	593	7632171
626	3608006	621	4959404	612	6310805	593	7662202
614	3638038	623	4989435	597	6340837	608	7692234
624	3668070	621	5019467	612	6370869	612	7722266

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
593	7752298	605	9103695	600	10455123	598	11806627
608	7782330	604	9133727	600	10485157	596	11836661
611	7812362	608	9163759	586	10515191	582	11866695
608	7842390	606	9193791	604	10545225	581	11896729
612	7872421	604	9223823	585	10575259	596	11926763
611	7902453	604	9253855	604	10605293	581	11956797
609	7932485	607	9283883	600	10635327	596	11986831
607	7962517	604	9313912	600	10665361	580	12016865
592	7992549	604	9343944	603	10695395	599	12046899
611	8022581	604	9373976	604	10725429	596	12076929
592	8052609	607	9404004	585	10755459	580	12106960
609	8082640	604	9434036	602	10785490	581	12136994
592	8112672	607	9464064	599	10815524	595	12167028
608	8142704	604	9494093	603	10845558	580	12197062
611	8172736	588	9524125	599	10875592	599	12227096
607	8202768	603	9554157	599	10905626	595	12257130
608	8232800	589	9584189	599	10935660	595	12287164
607	8262828	607	9614217	599	10965692	598	12317198
607	8292857	603	9644249	599	10995726	597	12347232
592	8322889	606	9674280	603	11025760	598	12377262
607	8352921	588	9704312	599	11055790	594	12407293
606	8382949	588	9734344	583	11085821	595	12437327
606	8412981	604	9764376	583	11115855	580	12467361
606	8443009	602	9794408	601	11145889	579	12497395
610	8473038	587	9824440	598	11175923	598	12527429
606	8503070	602	9854468	598	11205957	594	12557463
606	8533102	602	9884499	598	11235991	595	12587497
607	8563134	604	9914531	583	11266025	579	12617531
606	8593166	606	9944563	602	11296059	579	12647565
610	8623198	602	9974595	599	11326093	598	12677599
607	8653225	602	10004627	601	11356127	594	12707631
606	8683257	602	10034661	598	11386157	594	12737665
606	8713289	586	10064695	583	11416188	594	12767698
610	8743321	586	10094729	582	11446222	595	12797732
606	8773353	601	10124759	597	11476256	593	12827766
606	8803385	604	10154790	583	11506290	597	12857800
590	8833417	601	10184824	598	11536324	593	12887834
590	8863449	601	10214858	601	11566358	593	12917868
609	8893476	602	10244892	597	11596392	578	12947902
605	8923508	601	10274926	582	11626426	593	12977936
605	8953540	601	10304960	600	11656460	596	13007970
608	8983572	601	10334994	601	11686494	593	13038004
605	9013604	601	10365028	601	11716526	593	13068034
606	9043636	601	10395062	597	11746560	577	13098065
604	9073664	585	10425092	580	11776593	592	13128099

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
594	13158133	589	14509639	585	15861143	582	17212643
594	13188167	588	14539673	570	15891177	586	17242677
577	13218201	588	14569707	585	15921211	582	17272707
592	13248235	588	14599741	585	15951245	567	17302738
592	13278269	590	14629775	585	15981279	582	17332772
577	13308303	573	14659809	585	16011309	585	17362806
592	13338337	588	14689843	570	16041340	585	17392840
592	13368371	589	14719875	585	16071374	588	17422874
577	13398401	592	14749908	588	16101408	567	17452908
592	13428432	590	14779942	569	16131442	581	17482942
593	13458466	588	14809976	584	16161476	581	17512976
592	13488500	573	14840010	585	16191510	567	17543010
576	13518534	588	14870044	588	16221544	585	17573044
591	13548568	590	14900078	584	16251578	581	17603074
576	13578602	588	14930112	584	16281612	581	17633105
591	13608636	572	14960146	585	16311646	567	17663139
577	13638670	573	14990180	584	16341676	583	17693173
591	13668704	589	15020214	570	16371707	585	17723207
576	13698736	573	15050244	584	16401741	565	17753241
576	13728770	588	15080275	584	16431775	584	17783275
591	13758803	572	15110309	584	16461809	581	17813309
591	13788837	587	15140343	584	16491843	581	17843343
591	13818871	591	15170377	584	16521877	581	17873377
594	13848905	587	15200411	583	16551911	581	17903411
575	13878939	587	15230445	587	16581945	580	17933441
576	13908973	587	15260479	568	16611979	565	17963472
590	13939007	572	15290513	568	16642009	565	17993506
594	13969041	584	15320547	583	16672040	580	18023540
575	13999075	586	15350581	583	16702074	581	18053574
590	14029109	590	15380611	568	16732108	565	18083608
575	14059139	571	15410642	583	16762142	565	18113642
593	14089170	586	15440676	583	16792176	582	18143676
593	14119204	572	15470710	583	16822210	580	18173710
590	14149238	571	15500744	583	16852244	580	18203744
592	14179272	588	15530778	583	16882278	582	18233776
590	14209306	590	15560812	583	16912312	580	18263810
589	14239340	590	15590846	587	16942346	584	18293843
574	14269374	590	15620880	586	16972376	580	18323877
589	14299408	589	15650914	584	17002407	581	18353911
591	14329442	585	15680946	586	17032441	583	18383945
590	14359476	570	15710976	582	17062475	582	18413979
589	14389506	587	15741007	586	17092509	581	18444013
592	14419537	589	15771041	568	17122543	580	18474047
592	14449571	585	15801075	582	17152577	580	18504081
574	14479605	585	15831109	586	17182611	580	18534115

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
580	18564149	563	19915651	581	21267154	580	22618658
579	18594179	579	19945682	581	21297188	561	22648692
581	18624210	578	19975716	580	21327222	579	22678726
583	18654244	578	20005750	577	21357256	580	22708760
580	18684278	578	20035784	578	21387290	563	22738794
582	18714312	579	20065818	581	21417324	561	22768828
579	18744346	578	20095852	579	21447358	577	22798862
564	18774380	578	20125886	561	21477392	580	22828896
564	18804414	578	20155920	580	21507426	581	22858926
583	18834448	577	20185954	580	21537460	561	22888957
579	18864482	578	20215988	581	21567492	580	22918991
579	18894516	581	20246020	577	21597526	577	22949025
582	18924546	577	20276054	579	21627559	576	22979059
563	18954577	582	20306087	577	21657593	580	23009093
583	18984611	581	20336121	581	21687627	576	23039127
579	19014645	562	20366155	577	21717661	580	23069161
580	19044679	563	20396189	562	21747695	580	23099195
583	19074713	562	20426223	577	21777729	577	23129229
564	19104747	577	20456257	581	21807763	562	23159263
563	19134781	577	20486291	581	21837797	561	23189295
582	19164815	581	20516325	562	21867831	576	23219324
582	19194849	577	20546359	578	21897865	580	23249358
582	19224883	577	20576393	561	21927895	580	23279392
579	19254915	581	20606423	577	21957926	576	23309426
582	19284948	563	20636454	562	21987960	576	23339458
578	19314982	577	20666488	577	22017994	580	23369492
578	19345016	580	20696522	577	22048028	556	23399526
563	19375050	578	20726556	578	22078062	561	23429560
563	19405084	562	20756590	577	22108096	576	23459594
578	19435118	577	20786624	581	22138130	561	23489624
578	19465152	577	20816658	580	22168164	580	23519655
582	19495186	577	20846692	577	22198198	576	23549689
578	19525220	562	20876726	577	22228228	576	23579723
578	19555254	581	20906756	561	22258259	578	23609757
582	19585284	577	20936787	577	22288293	576	23639791
578	19615315	562	20966821	576	22318327	580	23669825
578	19645349	581	20996855	562	22348361	560	23699859
582	19675383	581	21026889	578	22378395	576	23729893
563	19705417	580	21056923	562	22408427	561	23759927
578	19735451	562	21086957	579	22438461	578	23789961
563	19765485	577	21116991	577	22468495	580	23819991
563	19795519	577	21147025	577	22498529	576	23850022
582	19825553	577	21177059	581	22528563	576	23880056
580	19855587	577	21207093	580	22558593	576	23910090
578	19885621	577	21237123	580	22588624	580	23940124

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
576	23970158	575	25321658	575	26673154	573	28024657
576	24000192	560	25351692	560	26703185	573	28054691
561	24030226	575	25381726	573	26733219	573	28084725
561	24060260	575	25411756	573	26763253	547	28114759
561	24090294	579	25441787	573	26793287	570	28144793
576	24120324	575	25471821	560	26823321	569	28174827
561	24150355	575	25501855	547	26853355	569	28204861
576	24180389	576	25531889	560	26883389	560	28234895
576	24210423	560	25561923	573	26913423	569	28264929
576	24240457	578	25591957	573	26943457	570	28294963
561	24270491	575	25621991	573	26973491	569	28324993
560	24300525	560	25652025	573	27003521	560	28355024
576	24330559	560	25682059	560	27033552	569	28385058
577	24360593	560	25712089	572	27063586	568	28415092
576	24390627	560	25742120	576	27093620	572	28445126
561	24420661	560	25772154	560	27123654	560	28475160
578	24450691	583	25802188	572	27153688	568	28505194
576	24480722	574	25832222	559	27183722	572	28535228
576	24510756	560	25862256	572	27213756	570	28565262
576	24540790	574	25892290	572	27243790	570	28595296
576	24570824	578	25922324	572	27273824	568	28625328
576	24600858	560	25952358	573	27303854	560	28655362
560	24630892	574	25982392	572	27333885	571	28685395
579	24660926	574	26012426	572	27363919	568	28715429
580	24690960	574	26042458	575	27393953	567	28745463
576	24720994	574	26072487	560	27423987	567	28775497
579	24751028	578	26102521	560	27454021	567	28805531
559	24781060	578	26132555	575	27484055	567	28835561
576	24811089	560	26162589	571	27514089	567	28865595
575	24841123	567	26192623	560	27544123	567	28895629
575	24871157	574	26222657	560	27574157	567	28925663
576	24901191	574	26252689	571	27604191	570	28955695
575	24931225	578	26282723	560	27634223	570	28985729
560	24961259	560	26312757	575	27664257	560	29015762
560	24991291	574	26342787	574	27694290	566	29045796
575	25021325	574	26372818	560	27724324	570	29075830
577	25051359	573	26402852	571	27754358	560	29105864
575	25081389	574	26432886	574	27784392	559	29135898
575	25111420	578	26462920	574	27814426	560	29165932
579	25141454	573	26492954	559	27844460	570	29195966
578	25171488	573	26522988	570	27874494	569	29226000
575	25201522	575	26553022	574	27904528	565	29256034
575	25231556	573	26583056	570	27934562	560	29286068
575	25261590	573	26613090	560	27964596	560	29316098
575	25291624	574	26643124	573	27994626	566	29346129

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
568	29376163	560	30727663	559	32079169	554	33430673
569	29406197	560	30757697	555	32109203	567	33460707
566	29436231	563	30787731	566	32139237	550	33490741
543	29466265	565	30817765	559	32169269	550	33520771
565	29496299	559	30847799	559	32199299	567	33550802
559	29526333	559	30877833	557	32229330	550	33580836
564	29556367	560	30907863	566	32259364	550	33610870
560	29586401	560	30937899	554	32289398	568	33640904
564	29616435	563	30967932	555	32319432	550	33670938
564	29646465	562	30997966	566	32349466	550	33700972
564	29676496	562	31028000	558	32379500	553	33731006
564	29706530	562	31058034	567	32409534	549	33761040
560	29736564	558	31088068	554	32439568	553	33791074
540	29766598	558	31118102	554	32469602	550	33821108
564	29796632	558	31148136	557	32499636	549	33851138
569	29826666	558	31178170	554	32529666	549	33881169
567	29856700	558	31208204	554	32559697	568	33911203
563	29886734	549	31238238	553	32589731	549	33941237
559	29916768	558	31268268	557	32619765	552	33971271
567	29946800	559	31298299	553	32649799	553	34001305
563	29976829	558	31328333	557	32679833	549	34031339
563	30006863	565	31358367	563	32709867	552	34061373
566	30036897	558	31388401	556	32739901	568	34091405
560	30066931	557	31418435	556	32769935	549	34121439
562	30096965	558	31448469	553	32799969	548	34151469
562	30126999	561	31478503	553	32830003	548	34181502
565	30157033	561	31508537	556	32860035	552	34211536
563	30187067	566	31538571	552	32890068	568	34241570
562	30217099	557	31568601	552	32920098	549	34271604
562	30247129	560	31598632	552	32950132	551	34301638
574	30277160	557	31628666	552	32980166	551	34331672
561	30307194	558	31658700	556	33010200	547	34361706
561	30337228	557	31688734	552	33040234	547	34391740
561	30367262	560	31718768	555	33070268	548	34421772
561	30397296	556	31748802	552	33100302	547	34451806
565	30427330	560	31778836	567	33130336	541	34481836
565	30457364	556	31808870	566	33160368	550	34511867
561	30487398	556	31838904	555	33190402	547	34541901
564	30517432	563	31868938	551	33220435	547	34571935
565	30547466	556	31898968	555	33250469	551	34601969
560	30577496	556	31928999	555	33280503	546	34632003
564	30607527	559	31959033	566	33310537	568	34662037
560	30637561	555	31989067	567	33340571	568	34692071
565	30667595	555	32019101	555	33370605	546	34722105
560	30697629	566	32049135	551	33400639	546	34752139

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
547	34782169	544	36133677	565	37485176	538	38836686
555	34812200	541	36163710	533	37515210	533	38866720
546	34842234	541	36193744	540	37545244	537	38896754
550	34872268	542	36223778	540	37575278	538	38926788
568	34902302	568	36253812	537	37605312	528	38956822
545	34932336	541	36283846	538	37635346	534	38986856
562	34962370	542	36313880	539	37665380	570	39016890
545	34992404	569	36343914	538	37695414	538	39046924
545	35022438	544	36373948	538	37725446	534	39076958
545	35052472	541	36403982	539	37755480	538	39106988
549	35082506	570	36434016	532	37785513	535	39137019
549	35112538	540	36464046	533	37815547	533	39167053
549	35142572	540	36494077	538	37845581	533	39197087
549	35172605	569	36524111	537	37875615	537	39227121
545	35202639	540	36554145	568	37905649	538	39257155
548	35232673	568	36584179	538	37935683	538	39287189
544	35262707	569	36614213	566	37965717	572	39317223
544	35292741	543	36644247	538	37995751	571	39347257
548	35322775	541	36674281	538	38025785	531	39377291
544	35352809	539	36704315	537	38055819	529	39407321
544	35382843	543	36734349	571	38085849	528	39437352
569	35412877	562	36764383	538	38115880	532	39467386
569	35442911	539	36794413	538	38145914	573	39497420
544	35472941	539	36824444	538	38175948	538	39527454
547	35502972	571	36854478	538	38205982	539	39557488
543	35533006	539	36884512	571	38236016	567	39587520
568	35563040	538	36914546	535	38266050	575	39617554
543	35593074	542	36944580	529	38296084	529	39647588
543	35623108	538	36974614	537	38326118	535	39677622
547	35653142	542	37004648	538	38356152	574	39707652
568	35683176	541	37034680	537	38386186	538	39737683
543	35713210	541	37064714	571	38416216	576	39767717
543	35743244	570	37094744	571	38446247	529	39797751
546	35773278	538	37124775	571	38476281	538	39827785
542	35803308	541	37154809	538	38506315	539	39857819
542	35833339	533	37184843	538	38536349	575	39887853
542	35863373	538	37214877	536	38566383	569	39917887
568	35893407	540	37244911	538	38596417	534	39947921
549	35923441	570	37274945	571	38626451	529	39977955
542	35953475	537	37304979	538	38656485	539	40007989
543	35983509	538	37335013	538	38686519	538	40038019
545	36013543	539	37365047	535	38716553	578	40068050
541	36043577	538	37395081	532	38746585	578	40098084
545	36073611	555	37425111	572	38776619	536	40128118
541	36103643	538	37455142	538	38806652	539	40158152

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
539	40188186	539	40398422	528	40608659	530	40818890
539	40218220	539	40428455	539	40638693	529	40848924
538	40248254	533	40458489	542	40668727	578	40878958
529	40278288	539	40488523	573	40698761	577	40908992
538	40308322	581	40518557	536	40728791	540	40939026
582	40338356	571	40548591	529	40758822		
539	40368388	530	40578625	539	40788856		

Tabla 7: Datos obtenido en la test del cálculo de descarga de la batería

4.2. Mínima duración de la batería

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
643	0	636	1802641	630	3604881	626	5407121
643	60450	636	1862715	630	3664955	626	5467195
643	120524	636	1922791	630	3725031	626	5527271
643	180598	635	1982865	630	3785105	626	5587345
642	240672	635	2042939	630	3845179	626	5647419
642	300748	635	2103015	630	3905255	626	5707495
641	360822	635	2163089	629	3965329	626	5767569
641	420896	635	2223163	629	4025403	625	5827643
641	480972	634	2283239	629	4085479	625	5887719
641	541046	634	2343313	629	4145553	625	5947793
640	601120	634	2403387	629	4205627	625	6007867
640	661198	634	2463463	629	4265703	625	6067945
640	721274	634	2523537	628	4325777	625	6128023
640	781348	633	2583611	628	4385851	625	6188103
639	841424	633	2643687	628	4445927	624	6248185
639	901498	633	2703761	628	4506001	624	6308263
639	961572	633	2763835	628	4566075	624	6368342
639	1021648	633	2823911	628	4626151	624	6428422
638	1081726	632	2883985	628	4686225	624	6488500
638	1141803	632	2944059	627	4746299	624	6548580
638	1201881	632	3004135	627	4806375	623	6608656
638	1261961	632	3064209	627	4866449	623	6668734
638	1322039	632	3124283	627	4926523	623	6728812
637	1382117	632	3184359	627	4986599	623	6788892
637	1442193	631	3244433	627	5046673	623	6848970
637	1502267	631	3304507	627	5106747	623	6909050
637	1562343	631	3364583	627	5166823	622	6969126
636	1622417	631	3424657	627	5226897	622	7029207
636	1682491	631	3484731	626	5286971	622	7089285
636	1742567	630	3544807	626	5347047	622	7149365

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
621	7209443	613	9912977	608	12616628	603	15320287
621	7269521	613	9973053	608	12676710	603	15380369
621	7329597	612	10033129	607	12736788	602	15440445
621	7389673	612	10093211	608	12796870	602	15500525
620	7449751	612	10153295	608	12856954	602	15560609
620	7509831	612	10213378	607	12917037	602	15620689
620	7569913	612	10273460	607	12977119	602	15680770
620	7629991	612	10333542	607	13037201	602	15740852
620	7690070	612	10393620	607	13097277	602	15800934
619	7750150	611	10453702	607	13157357	602	15861012
619	7810228	611	10513786	607	13217441	601	15921094
619	7870308	611	10573869	607	13277521	601	15981178
619	7930384	611	10633951	607	13337602	601	16041261
618	7990462	611	10694033	606	13397684	601	16101343
618	8050540	611	10754109	606	13457766	601	16161425
618	8110620	611	10814189	606	13517844	601	16221501
618	8170698	611	10874273	606	13577926	601	16281581
618	8230778	610	10934353	606	13638010	601	16341665
618	8290854	610	10994434	606	13698093	601	16401745
617	8350935	610	11054516	606	13758175	600	16461826
617	8411013	610	11114598	606	13818257	600	16521908
617	8471093	610	11174676	606	13878333	600	16581990
617	8531171	610	11234758	605	13938413	600	16642068
617	8591249	610	11294842	605	13998497	600	16702150
616	8651325	610	11354925	605	14058577	600	16762234
616	8711401	610	11415007	605	14118658	600	16822317
616	8771479	609	11475089	605	14178740	599	16882399
616	8831559	609	11535165	605	14238822	600	16942481
616	8891641	609	11595245	605	14298900	600	17002557
615	8951719	609	11655329	605	14358982	599	17062637
615	9011798	609	11715409	604	14419066	599	17122721
615	9071878	609	11775490	604	14479149	599	17182801
615	9131956	609	11835572	604	14539231	599	17242882
615	9192036	609	11895654	604	14599313	599	17302964
614	9252112	609	11955732	604	14659389	599	17363046
614	9312190	609	12015814	604	14719469	599	17423124
614	9372268	609	12075898	604	14779553	598	17483206
614	9432348	608	12135981	604	14839633	598	17543290
614	9492426	608	12196063	603	14899714	598	17603373
614	9552506	608	12256145	603	14959796	598	17663455
614	9612582	608	12316221	603	15019878	598	17723537
613	9672663	608	12376301	603	15079956	598	17783613
613	9732741	608	12436385	603	15140038	598	17843693
613	9792821	608	12496465	603	15200122	598	17903777
613	9852899	608	12556546	603	15260205	598	17963857

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
598	18023938	592	20727597	588	23431249	583	26134906
597	18084020	592	20787679	587	23491330	583	26194989
597	18144102	592	20847761	587	23551412	583	26255071
597	18204180	592	20907837	587	23611494	583	26315153
597	18264262	592	20967917	587	23671572	583	26375229
597	18324346	592	21028001	587	23731654	583	26435309
597	18384429	592	21088081	587	23791738	582	26495393
597	18444511	592	21148162	587	23851821	582	26555473
596	18504593	592	21208244	587	23911903	582	26615554
596	18564669	591	21268326	587	23971985	582	26675636
596	18624749	591	21328404	587	24032061	582	26735718
596	18684833	591	21388486	586	24092141	582	26795796
596	18744913	591	21448570	586	24152225	582	26855878
596	18804994	591	21508653	586	24212305	582	26915962
596	18865076	591	21568735	586	24272386	582	26976045
596	18925158	591	21628817	586	24332468	582	27036127
596	18985236	591	21688893	586	24392550	582	27096209
596	19045318	591	21748973	586	24452628	582	27156285
595	19105402	590	21809057	586	24512710	581	27216365
595	19165485	591	21869137	586	24572794	581	27276449
595	19225567	590	21929218	585	24632877	581	27336529
595	19285649	590	21989300	586	24692959	581	27396610
595	19345725	590	22049382	585	24753041	581	27456692
595	19405805	590	22109460	585	24813117	581	27516774
595	19465889	590	22169542	585	24873197	581	27576852
595	19525969	590	22229626	585	24933281	581	27636934
594	19586050	590	22289709	585	24993361	581	27697018
594	19646132	590	22349791	585	25053442	581	27757101
594	19706214	589	22409873	585	25113524	581	27817183
594	19766292	589	22469949	585	25173606	581	27877265
594	19826374	589	22530029	584	25233684	581	27937341
594	19886458	589	22590113	584	25293766	581	27997421
594	19946541	589	22650193	584	25353850	581	28057505
594	20006623	589	22710274	584	25413933	580	28117585
594	20066705	589	22770356	584	25474015	580	28177666
593	20126781	589	22830438	584	25534097	580	28237748
593	20186861	589	22890516	584	25594173	580	28297830
593	20246945	589	22950598	584	25654253	580	28357908
593	20307025	588	23010682	583	25714337	580	28417990
593	20367106	588	23070765	584	25774417	580	28478074
593	20427188	588	23130847	583	25834498	580	28538157
593	20487270	588	23190929	583	25894580	580	28598239
593	20547348	588	23251005	583	25954662	580	28658321
593	20607430	588	23311085	583	26014740	580	28718397
592	20667514	588	23371169	583	26074822	580	28778477

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
580	28838561	579	31542214	578	34245869	578	36949524
580	28898641	579	31602298	578	34305953	577	37009606
580	28958722	579	31662381	578	34366033	578	37069690
580	29018804	579	31722463	578	34426114	577	37129773
580	29078886	579	31782545	578	34486196	578	37189855
580	29138964	579	31842621	578	34546278	578	37249937
580	29199046	579	31902701	578	34606356	577	37310013
579	29259130	579	31962785	578	34666438	577	37370093
579	29319213	578	32022865	578	34726522	577	37430177
580	29379295	579	32082946	578	34786605	577	37490257
579	29439377	579	32143028	578	34846687	577	37550338
579	29499453	579	32203110	578	34906769	577	37610420
579	29559533	578	32263188	578	34966845	577	37670502
579	29619617	579	32323270	578	35026925	577	37730580
579	29679697	579	32383354	578	35087009	577	37790662
579	29739778	579	32443437	578	35147089	577	37850746
579	29799860	579	32503519	578	35207170	577	37910829
579	29859942	579	32563601	578	35267252	577	37970911
579	29920020	579	32623677	578	35327334	577	38030993
579	29980102	579	32683757	578	35387412	577	38091069
579	30040186	578	32743841	578	35447494	577	38151149
579	30100269	578	32803921	578	35507578	577	38211233
579	30160351	578	32864002	578	35567661	577	38271313
579	30220433	579	32924084	578	35627743	577	38331394
579	30280509	578	32984166	578	35687825	577	38391476
579	30340589	578	33044244	578	35747901	577	38451558
579	30400673	578	33104326	578	35807981	577	38511636
579	30460753	578	33164410	578	35868065	577	38571718
579	30520834	578	33224493	578	35928145	577	38631802
579	30580916	578	33284575	578	35988226	577	38691885
579	30640998	578	33344657	578	36048308	577	38751967
579	30701076	578	33404733	578	36108390	577	38812049
579	30761158	578	33464813	578	36168468	577	38872125
579	30821242	578	33524897	578	36228550	576	38932205
579	30881325	578	33584977	578	36288634	576	38992289
579	30941407	578	33645058	578	36348717	576	39052369
579	31001489	578	33705140	578	36408799	576	39112450
579	31061565	578	33765222	578	36468881	576	39172532
579	31121645	578	33825300	578	36528957	576	39232614
579	31181729	578	33885382	578	36589037	576	39292692
579	31241809	578	33945466	578	36649121	576	39352774
579	31301890	578	34005549	578	36709201	576	39412858
579	31361972	578	34065631	578	36769282	576	39472941
579	31422054	578	34125713	578	36829364	576	39533023
579	31482132	578	34185789	578	36889446	576	39593105

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
576	39653181	571	42357018	564	45060378	557	47763830
576	39713261	571	42417094	564	45120454	557	47823910
576	39773345	571	42477168	564	45180528	557	47883992
576	39833425	571	42537242	564	45240602	557	47944070
576	39893506	571	42597318	564	45300678	557	48004149
575	39953588	570	42657392	564	45360752	557	48064229
575	40013670	570	42717466	563	45420826	557	48124307
575	40073748	570	42777542	563	45480902	557	48184387
575	40133830	570	42837616	563	45540976	556	48244463
575	40193914	570	42897690	563	45601050	556	48304541
575	40253997	570	42957766	563	45661126	556	48364619
575	40314079	570	43017840	563	45721200	556	48424699
575	40374529	569	43077914	562	45781274	556	48484777
575	40434603	569	43137990	562	45841350	556	48544857
575	40494677	569	43198064	562	45901424	556	48604933
575	40554751	569	43258138	562	45961498	556	48665014
575	40614827	569	43318214	562	46021574	555	48725092
575	40674901	569	43378288	562	46081648	555	48785172
574	40734975	568	43438362	561	46141722	555	48845250
574	40795051	568	43498438	561	46201798	555	48905328
574	40855125	568	43558512	561	46261872	555	48965404
574	40915199	568	43618586	561	46321946	555	49025480
574	40975277	568	43678662	561	46382024	555	49085558
574	41035353	568	43738736	561	46442102	555	49145638
574	41095427	568	43798810	561	46502182	554	49205720
574	41155503	567	43858886	561	46562264	554	49265798
574	41215577	567	43918960	560	46622342	554	49325877
573	41275651	567	43979034	560	46682421	554	49385957
573	41335727	567	44039110	560	46742501	554	49446035
573	41395805	567	44099184	560	46802579	554	49506115
573	41455882	567	44159258	560	46862659	554	49566191
573	41515960	566	44219334	560	46922735	553	49626269
573	41576040	566	44279408	559	46982813	553	49686347
573	41636118	566	44339482	559	47042891	553	49746427
573	41696196	566	44399558	559	47102971	553	49806505
572	41756272	566	44459632	559	47163049	553	49866585
572	41816346	566	44519706	559	47223129	553	49926661
572	41876422	566	44579782	559	47283205	553	49986742
572	41936496	565	44639856	559	47343286	552	50046820
572	41996570	565	44699930	558	47403364	552	50106900
572	42056646	565	44760006	558	47463444	552	50166978
572	42116720	565	44820080	558	47523522	552	50227056
572	42176794	565	44880154	558	47583600	552	50287132
571	42236870	565	44940230	558	47643676	552	50347208
571	42296944	564	45000304	558	47703752	552	50407290

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
552	50467374	545	53171033	538	55874688	532	58578341
551	50527457	545	53231116	538	55934768	532	58638425
551	50587539	545	53291198	538	55994849	532	58698508
551	50647621	545	53351280	538	56054931	532	58758590
551	50707699	545	53411356	538	56115013	532	58818672
551	50767781	544	53471436	538	56175091	532	58878748
551	50827865	544	53531520	538	56235173	532	58938828
551	50887948	544	53591600	538	56295257	532	58998912
550	50948030	544	53651681	537	56355340	532	59058992
550	51008112	544	53711763	537	56415422	532	59119073
550	51068188	544	53771845	537	56475504	532	59179155
550	51128268	544	53831923	537	56535580	532	59239237
550	51188352	544	53892005	537	56595660	532	59299315
550	51248432	543	53952089	537	56655744	532	59359397
550	51308513	543	54012172	537	56715824	532	59419481
550	51368595	543	54072254	537	56775905	532	59479564
549	51428677	543	54132336	536	56835987	532	59539646
549	51488755	543	54192412	536	56896069	532	59599728
549	51548837	543	54252492	536	56956147	532	59659804
549	51608921	542	54312576	536	57016229	532	59719884
549	51669004	542	54372656	536	57076313	532	59779968
549	51729086	542	54432737	536	57136396	532	59840048
549	51789168	542	54492819	536	57196478	532	59900129
549	51849244	542	54552901	535	57256560	532	59960211
548	51909324	542	54612979	535	57316636	532	60020293
548	51969408	542	54673061	535	57376716	532	60080371
548	52029488	541	54733145	535	57436800	532	60140453
548	52089569	541	54793228	535	57496880	532	60200537
548	52149651	541	54853310	535	57556961	532	60260620
548	52209733	541	54913392	535	57617043	532	60320702
547	52269811	541	54973468	534	57677125	532	60380784
547	52329893	540	55033548	534	57737203	532	60440860
547	52389977	540	55093632	534	57797285	532	60500940
547	52450060	540	55153712	534	57857369	532	60561024
547	52510142	540	55213793	534	57917452	532	60621104
547	52570224	540	55273875	533	57977534	532	60681185
547	52630300	540	55333957	533	58037616	532	60741267
546	52690380	539	55394035	533	58097692	533	60801349
546	52750464	539	55454117	533	58157772	533	60861427
546	52810544	539	55514201	533	58217856	533	60921509
546	52870625	539	55574284	532	58277936	533	60981593
546	52930707	539	55634366	532	58338017	533	61041676
546	52990789	539	55694448	532	58398099	533	61101758
546	53050867	539	55754524	532	58458181	533	61161840
545	53110949	538	55814604	532	58518259	533	61221916

Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)	Lectura	Tiempo(ms)
533	61281996	533	61462241	533	61642483	534	61822732
533	61342080	533	61522323	534	61702565	534	61882814
533	61402160	533	61582405	534	61762649		

Tabla 8: Datos obtenido en la test del cálculo de descarga de la batería

A5. Código

5.1. Código del sistema alojado en el robot

```
#include <SoftwareSerial.h>

#include <XBee.h>


#define STATE0 0

#define STATE1 1

#define STATE2 2

#define STATE3 3

#define STATE4 4


//DEFINICIÓN DE TODAS LAS VARIABLES USADAS POR LA LIBRERIA xbee.h

XBee xbee = XBee();

int State = 1;

uint8_t StateSent[]={1};

XBeeResponse response = XBeeResponse();

Rx16Response rx16 = Rx16Response();

int address = 0xFFFF;

Tx16Request tx = Tx16Request(address,StateSent, sizeof(StateSent));

TxStatusResponse txStatus = TxStatusResponse();

SoftwareSerial mySerial(2, 3); // RX, TX


//RESTO DE VARIABLES USADAS

const int ReleOutB = 8;//Rele 1 Signal

const int ReleOutG = 6;//Rele 2 Signal

const int ReleOutR = 4;//Rele 2 Signal (EMERGENCIA)
```

```
const int NumberBut = 10;

const long interval = 500;

const int CARGAPROG = 11;


int ESTATECARGAPROG = 0;

int firstmlist=0;

int AdrresArray[10]={0,0,0,0,0,0,0,0,0,0}; //array de setas asociadas

int SwitchNumberConnected = 0;//numero de setas conectadas

int SwitchNumberConnectedAux = 0;

int count = 0;


//CONFIGURACIÓN DL MICROPROCESADOR

void setup() {


  pinMode(ReleOutB, OUTPUT);

  pinMode(ReleOutG, OUTPUT);

  pinMode(ReleOutR, OUTPUT);

  pinMode(CARGAPROG, INPUT);

  digitalWrite(CARGAPROG,HIGH);

  digitalWrite(ReleOutB, LOW);

  digitalWrite(ReleOutG, LOW);

  digitalWrite(ReleOutR, LOW);


  mySerial.begin(4800);

  xbee.begin(57600);

  mySerial.begin(4800);

}
```

```
void loop () {
 //INCIANCION DE LAS VARIABLES LOCALES
 int firstm = 0;
 int comp = 0;//varaiable para salir del bucle de asociacion de setas
 int newbutton = 0;
 int CTR = 0;
 long ReadTime = 0;
 long Tresponse = 0;
 int contj = 0;
 int comparison = 0;
 firstm = 0;
 comparison=0;
 comp = 0;//varaiable para salir del bucle de asociacion de setas
 newbutton = 0;
 address = 0xFFFF;
 SwitchNumberConnected=0;
 for (contj=0;contj<11;contj++){
  AdrresArray[contj]=0;
 }
 //Realización del BroadCAst
 Tresponse = millis();
 xbee.send(tx);
 xbee.send(tx);
 ReadTime = millis();
 //Escucahs de todas las setas y actuación sobre el robot
 while((millis()-ReadTime)<300){
```

```
   if (XBEERead(&State, &address, Tresponse)== 1){
     action(StateSent, State, &comparison);
    SwitchNumberConnected=LookFor(address, NumberBut,
&newbutton,SwitchNumberConnected);


   }
   else{
   }
  }
  digitalWrite(ReleOutB, LOW);
  digitalWrite(ReleOutG, LOW);
  digitalWrite(ReleOutR, LOW);
  //Comprobación de que todas las setas están conectadas
  if (comparison==0){
   if (firstmlist==0){
    SwitchNumberConnectedAux=SwitchNumberConnected;
    firstmlist=1;


   }
   else if((SwitchNumberConnectedAux) > (SwitchNumberConnected - newbutton)){
    emergency(StateSent);


   }
   else{
    SwitchNumberConnectedAux=SwitchNumberConnected;
   }
  }
}
```

```
//FUNCION QUE DECIDE QUE SE DEBE HACER EN FUNCION DEL ESTADO
void action( uint8_t *StateSentL,int StateL,int *compare){
 if (StateL == 5){
  emergency(StateSentL);
 }
 else if (StateL == 4){
  digitalWrite(ReleOutG,HIGH);
  digitalWrite(ReleOutB,HIGH);
  StateSent[0] = 2;
  *compare=1;
 }
 else if (StateL == 3){
  digitalWrite(ReleOutG,HIGH);
  *compare =1;
 }
 else if (StateL == 2){
  digitalWrite(ReleOutB,HIGH);
  *compare=1;
 }
 else{
  digitalWrite(ReleOutG,LOW);
  digitalWrite(ReleOutB,LOW);
 }
}
```

```
//FUNCION QUE BUSCA Y COMPLETA EL ARRAY DE DIRECCIONES
int LookFor(int AddrsL,  int lenght, int *NBut, int Conected){
  int count = 0;
  int result = 0;
  while ( count <= lenght){
    if (AdrresArray[count] == 0){
      AdrresArray[count] = AddrsL;
      *NBut ++;
      result = count+1;
      count = 11;

    }
    else if (AdrresArray[count] == AddrsL){
      result = Conected;
      count = 11;

    }
    else{
      count++;
    }
  }
  return result;
}
```

```
//FUNCION DE PARADA DE EMERGENCIA
void emergency(uint8_t *StateSentL){
  int i=0;
  digitalWrite(ReleOutB, LOW);
  digitalWrite(ReleOutG, LOW);
  digitalWrite(ReleOutR, HIGH);
  StateSentL[0]=1;
  firstmlist=0;
  for (i=0;i<10;i++){
    AdrresArray[i]= 0;
  }
  delay(1000);
  digitalWrite(ReleOutR, LOW);

}

//FUNCIÓN DE LECTURA DEL MÓDULO QUE LEE LOS PAQUETES
int XBEERead (int *dataL, int *addressL,long responsetime){
  long previousMillis = 0;
  int ErrorRead = 0;
  int ReturnError = 1;
  previousMillis = millis();
  //Lectura del módulo Xbee
  xbee.readPacket();
  if (xbee.getResponse().isAvailable()) {
```

```
   if (xbee.getResponse().getApiId() == RX_16_RESPONSE || xbee.getResponse().getApiId() ==
RX_64_RESPONSE){
     if (xbee.getResponse().getApiId() == RX_16_RESPONSE) {
       xbee.getResponse().getRx16Response(rx16);
       *addressL = rx16.getRemoteAddress16();
       *dataL = rx16.getData(0);
       ErrorRead=2;
       mySerial.println(*addressL);


     }
     else {
       ErrorRead=1;
       ReturnError = 0;
     }
   }
   else{
     ErrorRead=1;
     ReturnError = 0;
   }
 }
 else {
   ErrorRead=1;
   ReturnError = 0;
 }


 return (ReturnError);
}
```

```
//FUNCION QUE COMPRUEBA SI SE HA ESCRITO CORRECTAMENTE EN EL MODULO XBEE
char returnedOK () {
  // this function checks the response on the serial port to see if it was an "OK" or not
  char incomingChar[3];
  char okString[] = "OK";
  char result = 'n';
  int startTime = millis();
  while (millis() - startTime < 2000 && result == 'n') {   // use a timeout of 10 seconds
    if (Serial.available() > 1) {
      // read three incoming bytes which should be "O", "K", and a linefeed:
      for (int i=0; i<3; i++) {
        incomingChar[i] = Serial.read();
      }
      if ( strstr(incomingChar, okString) != NULL ) { // check to see if the respose is "OK"
       //if (incomingChar[0] == 'O' && incomingChar[1] == 'K') { // check to see if the first two
characters are "OK"
        result = 'T'; // return T if "OK" was the response
      }
      else {
        result = 'F'; // otherwise return F
      }
    }
  }
  return result;
}
```

5.2. Código del sistema alojado en la seta

```
#include <XBee.h>

#include <EEPROM.h>


#define LONG_INTERVAL_TIME 1500//Intervalo de tiempo de parpadeo del LED verde para busqueda de robot.

#define SHORT_INTERVAL_TIME 400//Intervalo de tiempo de parpadeo del LED cuando está conectado.

#define V1 593//Valor de la tension mas alta.

#define V2 574//Valor de la tension mas baja.


//Variables utilizadas para el funcionamiento de la libreria xbee.h:

XBee xbee = XBee();

uint8_t State = 0;

uint8_t StateSent[]={6};

XBeeResponse response = XBeeResponse();

Rx16Response rx16 = Rx16Response();

Tx16Request tx = Tx16Request(0X1,StateSent, sizeof(StateSent));

TxStatusResponse txStatus = TxStatusResponse();


//Selección de las entradas y salidas degitales:

const int ButtonConfi1 = 6;//Botón 1 de selección del robot.

const int ButtonConfi2 = 7;//Botón 2 de selección del robot.

const int ButtonConfi3 = 8;//Botón 3 de selección del robot.
```

```
const int ButtonActi1 = 4;//Button activacion 1

const int ButtonActi2 = 5;//Button activacion 2

const int Batery = 3;//Button de bateria

const int EmergencyStop = 2;//parada de emergencia

const int GreenLed = 9;//Led de State

const int BlueLed = 10;//Led de funcionamiento del robot

const int LedTension1 = 11;//Indicador de nivel de bateria 1

const int LedTension2 = 12;//Indicador de nivel de bateria 2

const int LedTension3 = 13;//Indicador de nivel de bateria 3

const int ConfEEPROM = 16;//jumper para la configuración del robot y escritura en la EEPROM

const int destidaddr = 1;

const int myaddresaddr = 0;


//Selección de la entrada analógica:

int TensionPin = A0;//Entrada de lectura de tension


//Declaración del resto de variables necesarias:

int EEPROMButton = 0;

int StateButton1 = 0;

int StateButton2 = 0;

int StateButton3 = 0;

int StateButtonActi1 = 0;

int StateButtonActi2 = 0;

int StateButtonEmergency = LOW;

int conected = 0;

long previousMillis = 0;

long interval = 0;
```

```
int GreenLedValue = LOW;
int aux=0;
int retardo=0;
int adrresmemory = 0;
byte myaddres = 0;
byte destid = 0;
//Función de configuración de entradas, salidas, velocidad de transmisión,
//selección de robot y selección del tiempo de retardo en el envio
void setup() {


  pinMode(ButtonConfi1, INPUT);
  pinMode(ButtonConfi2, INPUT);
  pinMode(ButtonConfi3, INPUT);
  pinMode(ButtonActi1, INPUT);
  pinMode(ButtonActi2, INPUT);
  pinMode(Batery, INPUT);
  pinMode(EmergencyStop,INPUT);
  pinMode(ConfEEPROM,INPUT);
  //Con estas sentencias se configuran las  entradas con las resistencias
  //de PULL-UP
  digitalWrite(ButtonConfi1,HIGH);
  digitalWrite(ButtonConfi2,HIGH);
  digitalWrite(ButtonConfi3,HIGH);
  digitalWrite(ButtonActi1,HIGH);
  digitalWrite(ButtonActi2,HIGH);
  digitalWrite(Batery,HIGH);
```

```
  digitalWrite(EmergencyStop,HIGH);
  digitalWrite(ConfEEPROM,HIGH);

  pinMode(GreenLed, OUTPUT);
  pinMode(BlueLed, OUTPUT);
  pinMode(LedTension1, OUTPUT);
  pinMode(LedTension2, OUTPUT);
  pinMode(LedTension3, OUTPUT);
  StateButton1 = digitalRead(ButtonConfi1);
  StateButton2 = digitalRead(ButtonConfi2);
  StateButton3 = digitalRead(ButtonConfi3);
  EEPROMButton = digitalRead(ConfEEPROM);
  destid = EEPROM.read(destidaddr);
  myaddres = EEPROM.read(myaddresaddr);
  if (EEPROMButton == LOW){
    cleanmemory();
    myaddres = readxbee();
    destid = writexbee(destid);
    writememory(myaddres,destid);
  }
  retardo = (myaddres -2) * 25;
  xbee.begin(57600);//configuracion XBEE
}

void loop () {
  //lectura del State desde el xbee
  XbeeReading();
```

```
aux=State;
switch (aux) {
   case 0:
      //espera
      if (conected == 0){
         interval = LONG_INTERVAL_TIME;
      }
      break;
    case 1:
      //seta asociada al robot
      conected == 1;
      interval = SHORT_INTERVAL_TIME;
      digitalWrite(BlueLed, LOW);
      break;
   case 2:
      //robot está en funcionamiento
      conected == 1;
      interval = SHORT_INTERVAL_TIME;
      digitalWrite(BlueLed, HIGH);
      break;
}
//Lectura de los botonos de funcionamiento
StateSent[0] = ButtonState();
if (aux != 0){
   xbee.send(tx);
}
```

```
  LighLED(&previousMillis, &interval);
  State = 0;
  aux=0;
  StateSent[0] = 0;
  if (digitalRead(Batery)== HIGH) {
    ReadBatery();
    LighLED(&previousMillis, &interval);
    //XbeeReading();
  }
  else{
    ResetBatery();
  }
  LighLED(&previousMillis, &interval);
}
```

```
//RESTO DE FUNCIONES IMPLEMENTADAS:

//FUNCION QUE COMPRUEBA SI SE HA ESCRITO CORRECTAMENTE EN EL MODULO XBEE:
char returnedOK () {
  // this function checks the response on the serial port to see if it was an "OK" or not
  char incomingChar[3];
  char okString[] = "OK";
  char result = 'n';
  int startTime = millis();
```

```
   while (millis() - startTime < 2000 && result == 'n') {   // use a timeout of 10 seconds
      if (Serial.available() > 1) {
       // read three incoming bytes which should be "O", "K", and a linefeed:
       for (int i=0; i<3; i++) {
       incomingChar[i] = Serial.read();
       }
       if ( strstr(incomingChar, okString) != NULL ) { // check to see if the respose is "OK"
       //if (incomingChar[0] == 'O' && incomingChar[1] == 'K') { // check to see if the first two
characters are "OK"
       result = 'T'; // return T if "OK" was the response
       }
       else {
       result = 'F'; // otherwise return F
       }
      }
  }
 return result;
}



//FUNCION QUE LE LOS BOTONES PULSADOS Y ACTUALIZA EL DATO QUE HAY QUE ENVIAR
uint8_t ButtonState(){


  uint8_t StateSentL = 1;
  StateButtonActi1 = digitalRead(ButtonActi1);
  StateButtonActi2 = digitalRead(ButtonActi2);
  StateButtonEmergency = digitalRead(EmergencyStop);
  //Botón de emergencia activado
```

```
  if (StateButtonEmergency == HIGH){
     StateSentL=5;
     aux=5;
  }
  //Botón de 1 y 2 activado
  else if (StateButtonActi1 == HIGH && StateButtonActi2 == HIGH){
     StateSentL=4;
     aux=4;
   }
   //Botón 1 activado
   else if (StateButtonActi1 == HIGH){
     StateSentL=3;
     aux=3;
   }
   //Botón 2 activado
   else if (StateButtonActi2 == HIGH){
        StateSentL=2;
        aux=2;
   }
   //ningún botón activado
   else{
        StateSentL=1;
        aux=0;
   }
  return(StateSentL);
}
```

```
//FUNCION QUE LEE LA TENSION DE LA BATERIA Y ENCIENDE LOS LEDS EN FUNCION DE LA TENSION
void ReadBatery(){
      int TensionValue = 0;//Valor de la tension
      TensionValue = analogRead(TensionPin);//Lectura de la tensión
      if (TensionValue > V1    ){
        digitalWrite(LedTension3, HIGH);
        digitalWrite(LedTension2, HIGH);
        digitalWrite(LedTension1, HIGH);
      }
      else if (TensionValue > V2){
        digitalWrite(LedTension3, HIGH);
        digitalWrite(LedTension2, HIGH);
      }
      else{
        digitalWrite(LedTension1, HIGH);
      }
}
```

```
//FUNCION QUE RESETEA LOS LEDS DE LA BATERIA
void ResetBatery(){
     digitalWrite(LedTension1, LOW);
     digitalWrite(LedTension2, LOW);
```

```
      digitalWrite(LedTension3, LOW);
}

//FUNCION QUE PERMITE EL PARPADEO DE LOS LEDS DE CONEXION
void LighLED(long *previousMillisL, long *intervalL){

 if (millis() - *previousMillisL > *intervalL) {
      *previousMillisL = millis();   // Recuerda la última vez que el State del LED cambió
      if (GreenLedValue == LOW){         // Si el LED está apagado lo enciende y viceversa.
        GreenLedValue = HIGH;
      }
      else{
        GreenLedValue = LOW;
      }
     digitalWrite(GreenLed, GreenLedValue);
  }
}



//Funcion que apaga los LEDs

void turnoff(){
   digitalWrite(LedTension1,LOW);
   digitalWrite(LedTension2,LOW);
   digitalWrite(LedTension3,LOW);
   digitalWrite(BlueLed,LOW);
   digitalWrite(GreenLed,LOW);
```

```
}



//FUNNCIÓN QUE LEE UN PAQUETE Y ENVIA LA RESPUESTA AL EMISOR
void XbeeReading(){
  xbee.readPacket();
   if (xbee.getResponse().isAvailable()) {
    // got something
      if (xbee.getResponse().getApiId() == RX_16_RESPONSE || xbee.getResponse().getApiId()
== RX_64_RESPONSE) {
        // got a rx packet
          if (xbee.getResponse().getApiId() == RX_16_RESPONSE) {
             xbee.getResponse().getRx16Response(rx16);
             State = rx16.getData(0);
             aux=State;
             StateSent[0] = 1;
             delay(retardo);
             xbee.send(tx);
             LighLED(&previousMillis, &interval);
          }
          else{
             State=0;
          }
        }
        else{
           State=0;
        }
  }
```

```
    else{
        State=0;
    }
}

//FUNCIÓN QUE BORRA LA MEMORIA
void cleanmemory(){
    for (int i = 0; i < 512; i++){
        EEPROM.write(i, 0);
    }
    digitalWrite(BlueLed,HIGH);
}



//FUNCION QUE LEE LA DIRECCION DE LA SETA
byte readxbee(){
    byte result = 0;
    int waitr = 0;
    long timerread = 0;
    Serial.begin(57600);
    digitalWrite(LedTension3,HIGH);
    Serial.print("+++");
    delay(1100);
    // wait for a response from the XBee for 2000 ms, or start
    // over with setup if no valid response comes
    if (returnedOK() == 'T') {
     // if an OK was received then continue
```

```
  }
  else {
     setup(); // otherwise go back and try setup again
  }

  //Selección del tiempo de retardo:
  Serial.println("\r\nATMY");
  timerread = millis();
  while (millis() - timerread < 2000 && waitr == 0) {   // use a timeout of 10 seconds
        if (Serial.available() > 1) {
           result = (Serial.read()-48);
           if (result == 1) {
              result = (Serial.read()-38);
           }
           waitr = 1;
        }
  }
  return (result);
}


//FUNCION QUE ESTABLECE A QUE ROBOT TE QUIERES CONECTAR
byte writexbee(byte prevaddres){
  byte result;
  //Selección del robot:
  digitalWrite(LedTension2,HIGH);
  if (StateButton1 == HIGH && StateButton2 == HIGH && StateButton3 == HIGH &&
prevaddres != 0){
```

```
      Serial.println("\r\nATID3330");
      result = 0;
      delay(1100);
      if (returnedOK() == 'T') {
     // if an OK was received then continue
      }
      else {
         turnoff();
         setup();
      }

   }
   else if (StateButton1 == HIGH && StateButton2 == HIGH && StateButton3 == LOW &&
prevaddres != 1){
      Serial.println("\r\nATID3331");
      result = 1;
      delay(1100);
      if (returnedOK() == 'T') {
          // if an OK was received then continue
      }
      else{
          turnoff();
          setup();
      }
   }
   else if (StateButton1 == HIGH && StateButton2 == LOW && StateButton3 == HIGH &&
prevaddres != 2){
```

```
      Serial.println("\r\nATID3332");
      result = 2;
      delay(1100);
      if (returnedOK() == 'T') {
          // if an OK was received then continue
      }
      else{
         turnoff();
         setup();
      }
   }
   else if (StateButton3 == HIGH && StateButton2 == LOW && StateButton1 == LOW &&
prevaddres != 3){
      Serial.println("\r\nATID3333");
      result = 3;
      delay(1100);
      if (returnedOK() == 'T') {
          // if an OK was received then continue
      }
      else{
         turnoff();
         setup();
      }
   }
   else if (StateButton1 == LOW && StateButton2 == HIGH && StateButton3 == HIGH &&
prevaddres != 4){
      Serial.println("\r\nATID3334");
      result = 4;
```

```
      delay(1100);
      if (returnedOK() == 'T') {
          // if an OK was received then continue
      }
      else{
          turnoff();
          setup();
      }
   }
   else if (StateButton1 == LOW && StateButton2 == HIGH && StateButton3 == LOW &&
prevaddres != 5){
      Serial.println("\r\nATID3335");
      result = 5;
      delay(1100);
      if (returnedOK() == 'T') {
          // if an OK was received then continue
      }
      else{
          turnoff();
          setup();
      }
   }
   else if (StateButton1 == LOW && StateButton2 == LOW && StateButton3 == HIGH &&
prevaddres != 6){
      Serial.println("\r\nATID3336");
      result = 6;
      delay(1100);
```

```
    if (returnedOK() == 'T') {
        // if an OK was received then continue
    }
    else{
        turnoff();
        setup();
    }
  }
  else if (StateButton1 == LOW && StateButton2 == LOW && StateButton3 == LOW &&
prevaddres != 7){
    Serial.println("\r\nATID3337");
    result = 7;
    delay(1100);
    if (returnedOK() == 'T') {
        // if an OK was received then continue
    }
    else{
        turnoff();
        setup();
    }
  }
  else{
    result = prevaddres;
  }
  digitalWrite(LedTension1,HIGH);
  Serial.println("\r\nATWR");
  delay(1100);
  if (returnedOK() == 'T') {
```

```
    // if an OK was received then continue
  }
  else {
    turnoff();
    setup();
  }
  Serial.println("\r\nATCN");
  if (returnedOK() == 'T') {
    // if an OK was received then continue
  }
  else {
    turnoff();
    setup();
  }
  return (result);
}

//FUNCION QUE ESCRIBE LOS PARÁMETROS EN LA MEMORIA
void writememory(byte myaddresL,byte destidL){
  EEPROM.write(myaddresaddr, myaddresL);
  EEPROM.write(destidaddr, destidL);
  turnoff();
  while (1){
    digitalWrite(GreenLed,HIGH);
  }
}
```